VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Využití systému Maple a programu Java View k vizualizaci problematiky dvojných a trojných integrálů

Using Maple and Java View to Visualize Double and Triple Integrals

2009

Vojtěch Milota

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 3. května 2009 ....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla, zvláště pak mé vedoucí bakalářské práce RNDr. Petře Šarmanové, Ph.D. .

Abstrakt

Cílem této bakalářské práce je vytvořit webovou galerii příkladů s interaktivní 3D grafikou vytvořenou za pomoci systému Maple a programu JavaView pro podporu výuky problematiky dvojných a trojných integrálů. Takto vytvořená a okomentovaná grafika přispívá k pochopení nových pojmů a k objasnění geometrického významu řešených úloh. Práce zahrnuje výběr vhodných příkladů, u kterých jsou vytvořeny grafické objekty systémem Maple. Tato interaktivní grafika z Maplu je vyexportována pomocí knihovny JavaViewLib a následně prezentována na webu pomocí programu JavaView, který všechnu interaktivitu grafických objektů zachovává.

Klíčová slova: Maple, Calcplot, ASU 10, JavaView, HTML stránky, jsMath, export grafů

Abstract

The main goal of the Bachelor thesis is to create the web gallery colleciton of examples with interactive 3D graphic objects created with the help of the Maple software and computer programme JavaView to support the educational problems such as double and triple integrals. The created and commented graphic makes it easier to understand new conceptions and also to make the outcome of geometric problem solution clear. This work includes the selection of useful examples which are created by Maple software. This interactive graphic from Maple is exported with the help of JavaViewLib library and afterwards presented on the World Wide Web with the help of the computer programme JavaView which keeps all the interactivity of those graphic objects.

Keywords: Maple, Calcplot, ASU 10, JavaView, HTML pages, jsMath, graphs export

Seznam použitých zkratek a symbolů

CSS	–	Cascading Style Sheets
JVL	–	Java View Library
PHP	–	Hypertext Preprocessor
XHTML	–	Extensible Hyper Text Markup Language

Obsah

Seznam tabulek

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

V dnešní době se stává pro mnohé studenty samozřejmostí vyhledávání řešení studijních i jiných problému na internetu. Je pravda, že internet je skoro bezedná studna, leč v některých oborech stále zaostává. Jedním z těchto vědomostních oborů je i paradoxně matematika, jedna z nejzákladnějších vědních disciplín. Při snaze naleznout potřebnou teorii v českém jazyce se nemálo z nás setkalo s neúspěchem. Vždyť stále ve 21. století pocházejí jedny z nejlepších knih zahrnující skoro celé spektrum matematická analýzy, tím myslím Matematiku 1 a Matematiku 2 autorů Charvát a spol., z roku 1993 a v současné době jsou stále ještě hojně využívány při výuce. Během studia matematiky jsem se většinou setkal s naskenovanými poznámkami umístěnými na serveru školy, kde je nikdo jiný nemohl najít. Přitom elektronické publikace jsou snadněji aktualizovatelné, odpadají náklady na výtisk nových vydání, jsou rychle rozšiřitelné a hlavně přístup k nim je okamžitý, bez nutnosti chodit do knihovny. V dnešní době rozpuku informačních technologií nabírají elektronické dokumenty zcela nových rozměrů. Jsou vytvořeny programovací standardy jež umožňují propojit mnohé aplikace a vytvářejí se tak více a více interaktivnější dokumenty. Pro koncového uživatele jsou tak vytvořeny dokumenty s dynamickými prvky jako jsou applety nebo animace vytvořené programem Flash. Tím jsou 2D obrázky nahrazeny 3D objekty s nimiž je možné nadále pracovat. Takto vylepšené dokumenty mají největší uplatnění ve vědních oborech matematiky jako je geometrie nebo matematická analýza, kde je vyžadován určitý stupeň představivosti. Právě kde představivost selhává, síla těchto dokumentů začíná.

Tato bakalářská práce popisuje vytvoření sbírky příkladů týkající se problematiky dvojných a trojných integrálů prezentovaných na webu. Je v ní popsána i metodika matematické sazby na webu za použití java scriptu jsMath. Tento java script převádí latexovský kód matematického zápisu do výstupu s fonty přibližující se kvalitou výstupu matematiky z latexu samotného. Tento způsob tak nahrazuje ruční převod matematických symbolů do obrázků. Hlavním cílem je mimo jiné vizualizace dvojných a trojných integrálů při zachování jejich interaktivity. S programy Maple a JavaView bylo dosaženo požadovaných výsledků. Vyexportované grafy jsou umístěné na web prostřednictvím java appletu při uchování interaktivity. V dnešní době je java rozhraní implementováno téměř do každého webového prohlížeče a odpadají tak nároky na uživatele.

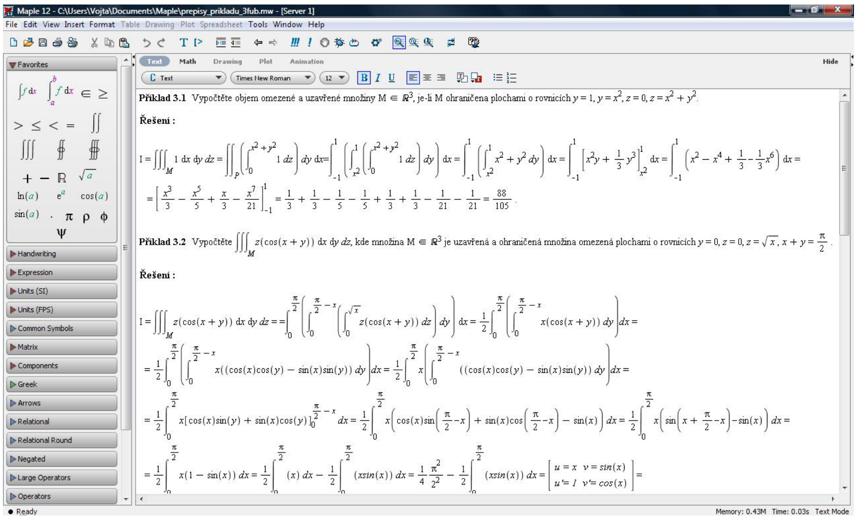
Předmětem následujícího textu je popis programu Maple a jeho možnosti vytváření grafů pomocí knihovny Calcplot, dále propojení programu JavaView s programem Maple a vytváření grafů a v neposlední řadě tvorby internetových stránek s applety a matematickým textem za použití scriptu jsMath. Těžištěm bakalářské práce je popsání vytvoření stránek se všemi jejími prvky. Stránky jsou dostupné na adrese `http://am.vsb.cz/milota` nebo na CD přiloženém k této práci.

Vlastní text práce je rozdělen do šesti kapitol dle probírané problematiky. Druhá kapitola se zabývá programem Maple, seznámením a nastavením jednotlivých funkcí programu využitých v práci. Třetí kapitola se věnuje balíčku Calcplot, jenž přidává další funkce ke snadné vizualizaci dvojných a trojných integrálů. Následující kapitola se věnuje hlavnímu cíli práce, což je propojení programu Maple s programem JavaView. Další dvě kapitoly popisují tvorbu webových stránek s nastavením všech potřebných prvků zobrazení jako jsou applety a matematický text pomocí scriptu jsMath. V poslední kapitole jsou uvedeny ukázky příkladů, které jsou dostupné na webu.

2 Maple

Maple 12 je nástroj pro vývojáře, výzkumníky i učitele v každé matematické nebo technické disciplíně. Je postaven na světově uznávaném výpočetním jádru, které poskytuje rozsáhlé možnosti výpočtů pro každou oblast matematiky, pro výuku matematiky a věd souvisejících. Učitelé mohou použít při výuce mnohem komplikovanější problémy týkající se přímo života kolem nás a výzkumníci mohou vytvářet mnohem sofistikovanější a komplexnější algoritmy a modely. Zkoumání, vizualizace a řešení komplexních matematických úloh je se systémem Maple 12 mnohem jednodušší, názornější, rychlejší a v neposlední řadě díky Rich Technical Document i řádně zdokumentovatelné. Díky chytrému technickému prostředí poskytuje "klikací"nástroje, které prakticky odstraní křivku osvojování znalostí pro vykonávání sofistikovanějších úkolů. Maple 12 také poskytuje nástroje pro produkování dokumentů, které jsou plně interaktivní a vypadající jako profesionální učebnice. Dále Maple 12 přináší mnoho novinek jako například zdokonalenou podporu výuky s novými balíčky pro studenty nebo nové grafické možnosti dramaticky zjednodušující vytváření komplexních grafů.

Maple 12 opravdu doznal oproti mnou dříve používané verzi 10 řadu vylepšení. To se týče zejména grafického „kabátku", které bylo jistě upraveno tak, aby korespondovalo s novou tváří Windows Vista jednak graficky a jednak kompatibilitou, která byla v Maplu 10 i přes instalaci v kompatibilním módu poruchová. Maple 12 je tak plně funkční pod tímto operačním systémem a nesetkal jsem se s vážnějšími problémy, až na celkem vysoké zatížení procesoru a paměti při práci s vícero 3D grafy, což mě u výkonného procesoru Core2Duo, 2 GB RAM paměti a výkonné grafické karty zarazilo. Naopak příjemnou změnou je takzvaný „document mode", ve kterém se příkazy nezadávají přes zobáček $>$ z „worksheet mode", ale přímým vypsáním funkce, což opravdu umožňuje vytvářet pěkné a přehledné dokumenty profesionální kvality.



Obrázek 1: Náhled Maple prostředí

2.1 Funkce plot a její nastavení

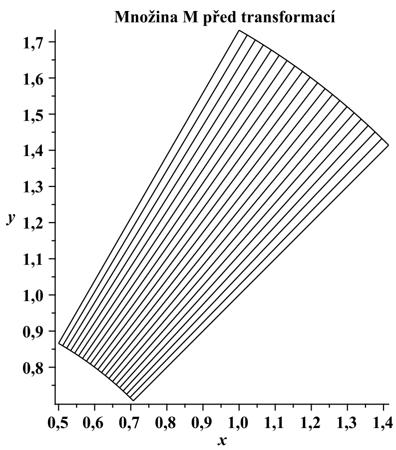
Při vizualizaci problematiky dvojných integrálů využíváme mapleovskou funkci plot, která vytváří dvojrozměrné grafy. Funkce plot se volá příkazem plot(f, x=a..b), kde f je požadovaná funkce, a a b jsou intervaly, na kterých má být zobrazena. Volby uvedené níže můžou být použity u příkazů, které vytváří 2D grafy.

- axes=f, specifikuje typ os, můžeme si vybrat : boxed, frame, none, nebo normal.
- axesfont=l, určuje typ fontu u značek os, zadává se stejně jako typ fontu, viz nastavení font.
- axiscoordinates=t, dává na výběr souřadnicový systém při zobrazení os. Hodnota t může nabývat hodnot polar nebo cartesian. Standardně je volen kartézský systém os. Jestliže nastavíme t na polar, potom zvolíme polární systém os. Toto nastavení je používáno spolu s coords=polar nastavením.
- caption=c, udává popis grafu. Hodnota c může nabývat jakéhokoliv výrazu.
- color=n nebo colour=n, umožňuje nastavit barvu křivky.
- coords=cname, dává na výběr souřadnicový systém. Je na výběr cartesian nebo polar. K vykreslení polární křivky v polárním systému používáme axiscoordinates=polar spolu s coords=polar nastavením.
- filled=truefalse, jestliže nastavíme na true, oblast mezi křivkou a osou x je vyplněna barvou.
- font=l, nastavuje font textu u grafu, kde hodnota l je [family, style, size]. Hodnota family je jedna z TIMES, COURIER, HELVETICA, nebo SYMBOL. Hodnoty musí být nastaveny velkými písmeny .Pro TIMES styl můžeme vybrat druh písma ROMAN, BOLD, ITALIC, nebo BOLDITALIC.Pro styl HELVETICA a COURIER máme na výběr z BOLD, OBLIQUE, nebo BOLDOBLIQUE. Nastavení velikosti písma je dáno číselně.
- labels=[x, y], toto nastavení určuje popisek pro osy.
- labeldirections=[x, y], udává směr popisků u os. X a y mohou nabývat hodnot ”horizontal”nebo ”vertical”. Standardně je nastaveno horizontal.
- labelfont=l, specifikuje font u popisku os, zadává se stejně jako typ fontu.
- linestyle=t, udává styl křivky. Hodnota t může být : solid, dot, dash, dashdot, longdash, spacedash, spacedot. Výchozí nastavení je solid.
- scaling=s, při vykreslení dává na výběr zachování měřítka 1:1. Na výběr je constrained nebo unconstrained. Standartně je voleno unconstrained, což znamená nezachovávat měřítko.

- thickness=n, specifikuje šířku čáry funkce v grafu. Výchozí nastavení je 0.
- title=t, udává název grafu. Hodnota t může nabývat jakéhokoliv výrazu.
- titlefont=l, specifikuje font u názvu, zadává se stejně jako typ fontu.
- transparency=t, určuje průhlednost křivky funkce. Hodnota t musí být číslo v rozmezí 0 až 1, kdy 0 znamená neprůhledné a 1 znamená absolutně průhledné.
- view=[xmin..xmax, ymin..ymax], toto nastavení udává rozmezí zobrazení křivky. Standardně je vykreslena celá křivka

Příklady použití[1] :

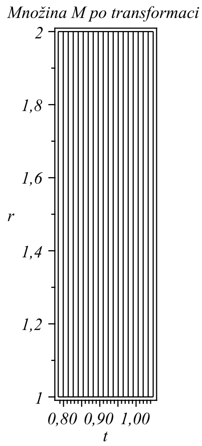
```
>  drdtplot(r = 1..2, theta = (1/3)*Pi..(1/4)*Pi, labels = [x, y], axes
= normal, title = "Množina M před transformací", color = black, scaling
= constrained, font = [TIMES, BOLD, 12]);
```



Obrázek 2: Nastavení mapleovské funkce plot u grafu funkce drdtplot

```
>  dydxplot(y = 1..2, x = (1/4)*Pi..(1/3)*Pi, labels = [t, r], axes =
BOXED, title = "Množina M po transformaci", color = black, scaling=constrained,
font = [COURIER, OBLIQUE, 12]);
```

[1]Funkce balíčku Calcplot, viz následující kapitola



Obrázek 3: Nastavení mapleovské funkce plot u grafu funkce dydxplot

2.2 Funkce plot3d a její nastavení

Při vizualizaci problematiky trojných integrálů využíváme mapleovskou funkci plot3d, která vytváří 3-D grafy. Funkce plot3d se volá příkazem plot3d(f, a..b, c..d, options), kde f je požadovaná funkce, a a b, c a d jsou intervaly, na kterých má být zobrazena a nastavení options je uvedeno níže. Všechny nastavení pro funkci plot jsou použitelné i pro funkci plot3d a proto jsou vypsané ta nastavení, která se týkají pouze plot3d.

- ambientlight=[r, g, b], určuje červenou, zelenou a modrou složku intenzity ambient světla pro požadované nastavení. Hodnoty r, g, b musí být číslo v rozmezí 0 až 1.

- glossiness=g, specifikuje lesk vykresleného povrchu. Hodnota lesku g, která určuje množství světla odraženého od povrchu musí být číslo v rozmezí 0 až 1, kde hodnota 0 znamená matný povrch neodrážející světlo a hodnota 1 lesklý povrch s maximálním odrazem světla. Standardní nastavení je 0. Odraz světla může být vyrenderován v případě zapnutí alespoň jednoho zdroje světla.

- light=[phi, theta, r, g, b], přidává přímý zdroj světla ze směru phi, theta, což jsou úhly ve sférických souřadnicích s červenou, zelenou a modrou intenzitou udávanou r, g, b. Hodnoty r, g, b musí být číslo v rozmezí 0 až 1.

- lightmodel=x, nastavuje předdefinovaný světelný model k osvětlení povrchu tělesa. Máme na výběr z none, light1, light2, light3, a light4. Hodnota musí být napsána malým písmem.

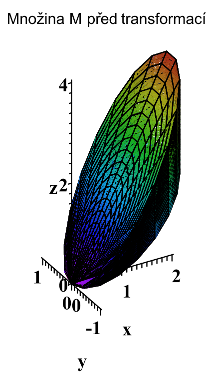
- orientation=[theta, phi], udává úhel pohledu, z kterého je těleso zobrazeno. Úhel pohledu je popsán úhly theta a phi ve sférických souřadnicích, kde výchozí nastavení pro theta a phi je 45 stupňů.

- projection=r, specifikuje perspektivu z které je těleso viděno, kde r je číslo v rozmezí 0 až 1. 1 značí ortogonální projekci a 0 znamená širokoúhlou perspektivu, r může také nabývat jmen : fisheye, normal nebo orthogonal, které korespondují s hodnotami 0, 0.5 a 1. Výchozí nastavení je orthogonální.

- shading=s, určuje zbarvení povrchu, kde s je : xyz, xy, z, zgrayscale, zhue nebo none. Standardně záleží na stínovaném tělesu.
- style=s, styl grafu musí být jeden z následujících surface, surfacewireframe, contour, surfacecontour, wireframe, wireframeopaque nebo point. Výchozí nastavení je surfacewireframe.

Příklady použití[2] :

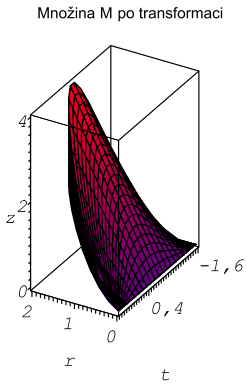
```
> dzdrdtplot(z = r*cos(theta)..2*r*cos(theta), r = 0..2*cos(theta), theta
= -(1/2)*Pi..(1/2)*Pi, axes = normal, labels = [x, y, z], style = surface-
wireframe, lightmodel = light2, glossiness = 0.8, shading = zhue, orientation
= [240, 60], scaling = constrained, font = [TIMES, BOLD, 12], title =
"Množina M před transformací");
```



Obrázek 4: Nastavení mapleovské funkce plot3d u grafu funkce dzdrdtplot

```
> dzdxdyplot(z = x*cos(y)..2*x*cos(y), x = 0..2*cos(y), y = -(1/2)*Pi
.. (1/2)*Pi, axes = boxed, labels = [r, t, z], style = surfacewireframe,
lightmodel = light3, glossiness = .8, shading = z, orientation = [120,
60], scaling = constrained, title = "Množina M po transformaci", font
= [COURIER, OBLIQUE, 12]);
```

[2]Funkce balíčku Calcplot, viz následující kapitola



Obrázek 5: Nastavení mapleovské funkce plot3d u grafu funkce dzdydxplot

2.3 Balíček Student-MultivariateCalculus

Balíček Student-MultivariateCalculus je navržený k tomu, aby pomohl učitelům prezentovat a studentům porozumět základním poznatkům ze standardního kurzu funkcí více proměnných. V mém případě jsem ho použil z důvodu kontroly výsledků počítaných integrálů, popřípadě k zobrazení aproximace integrované funkce. Dobré uplatnění předpokládám i co se týče podpory výuky. Balíček obsahuje dvě hlavní součásti : interaktivní a vizualizační.

Vizualizační součásti jsou navrženy k tomu, aby pomáhali při porozumění základního postupu výpočtu a pochopení teorie funkcí více proměnných. Tyto rutiny standardně produkují grafy a většina z nich může vrátit jedno nebo více reprezentací dané problematiky. Můžeme si dle uvážení vybrat jak bude graf vygenerovaný pomocí těchto rutin vypadat. Vyobrazení každého objektu v grafu může být pozměněno užitím nastavení dané procedury.

Vizualizační příkazy jsou : ApproximateInt, CenterOfMass, ChangeOfVariables, CrossSection, DirectionalDerivative, FunctionAverage, Gradient, Jacobian, LagrangeMultipliers, MultiInt, SecondDerivativeTest, SurfaceArea, TaylorApproximation.

Interaktivní součásti využívají technologie Maplet[3] k tomu, aby nám pomohla pochopit základní teorii kurzu funkcí více proměnných. Tyto příkazy zobrazí jedno nebo více dialogových oken dovolující upravovat funkci a měnit různé volby grafu.

Interaktivní příkazy jsou : ApproximateIntTutor, CrossSectionTutor, DirectionalDerivativeTutor, GradientTutor, TaylorApproximationTutor.

2.3.0.1 ApproximateInt

K pochopení problematiky integrálního počtu funkcí více proměnných můžeme využít funkci ApproximateInt. Tu lze sice použít pouze pro dvojné integrály, ale i přesto je graf názorný, a pomůže studentům pochopit danou problematiku. Příkaz ApproximateInt(f(x,y),x=a..b,y=c..d, opts) vrací graf aproximace integrované funkce f(x,y), kde x,y jsou jména nezávislých proměnných, a,b,c,d jsou integrační meze.

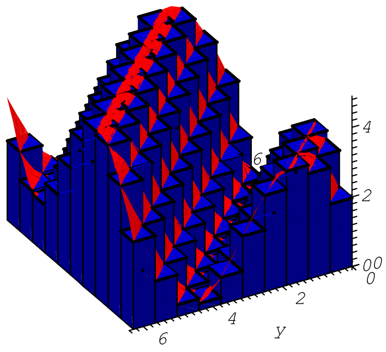
[3]Maplet v rámci systému poskytuje interaktivní uživatelské prostředí v podobě dialogových oken postavených na technologii Java.

Nastavení opts, které mění požadované hodnoty, může být jakýkoliv výraz z následujícího výpisu.

- coordinates = cartesian nebo polar, určuje zda je integrál počítán v kartézských nebo polárních souřadnicíh. V polárních souřadnicích první proměnná označuje poloměrovou složku.
- frames = a..b, specifikuje počet snímků v animaci, b-a+1.
- functionoptions = list , specifikuje nastavení funkce f(x,y) v grafu, je stejné jako u plo3d.
- method = midpoint, upper, lower, nebo random, počítá aproximaci integrálu pomocí midpoint, upper, lower, nebo random Riemannových součtů. Výchozí nastavení je midpoint.
- output = value, plot, nebo animation, nastavuje vracející hodnotu na value, která vrací hodnotu aproximace a graf není vykreslen, plot, která vrací hodnotu aproximace a graf nebo animation, která vrací animaci.
- partition = [a, b], nastavuje dělení intervalu na daných mezích. Standardně je [5, 5].
- prismoptions = list, udává další nastavení aproximačních dílů v grafu, jako barva a atd.. Nastavení je stejné jako u plo3d.
- showfunction = true nebo false, určuje zda je funkce f(x,y) vykreslena. Výchozí nastavení je true.
- title = string, udává název grafu.

Příklad :

```
>  with(Student[MultivariateCalculus]);
>  ApproximateInt(sqrt(x+y)+2*sin(x+y), x = 0..2*Pi, y = 0..2*Pi, method
= midpoint, coordinates = cartesian, partition = [10, 10], output = plot,
scaling = constrained, axes = normal, lightmodel = light3, shading = z,
orientation = [150, 60], scaling = constrained, title = "", font = [COURIER,
OBLIQUE, 12]);
```



Obrázek 6: Graf vygenerovaný pomocí funkce ApproximateInt

2.3.0.2 MultiInt

Další užitečná funkce při počítání integrálů funkcí více proměnných je MultiInt. Tato funkce nahradila dříve známé int, Int, DoubleInt a TripleInt. Příkaz MultiInt(f(x,y,z), x=a..b, y=c..d, z=e..f, opts) vrací hodnotu integrálu funkce f(x,y,z) na mezích x=a..b, y=c..d, z=e..f. Nastavení opts mění požadovaný výstup.

- coordinates = cartesian[x,y], polar[r,theta], cartesian[x,y,z], cylindrical[r,theta,z], spherical[r,phi,theta], určuje souřadnicový systém, který má být použit.
- output = value, integral nebo steps, toto nastavení kontroluje výstup. Value znamená, že na výstupu je výsledek daného intergálu. Hodnota integral vrací na výstup integrál tak jak je zadán. Hodnota steps vrací kroky postupné integrace.

Příklad :

```
> with(Student[MultivariateCalculus]);
> MultiInt(x+y^2, y = x^2..sqrt(x), x = 0..1, output = steps);
```

2.4 Knihovna ASU 10

Původními autory knihovny ASU jsou Shannon Holland a Matthias Kawski. Knihovna byla následně upravena mou vedoucí práce RNDr. Petrou Šarmanovou, Ph.D. tak, aby byla funkční pro Maple 10 (je funkční i pro Maple 12). Tímto způsobem vznikla knihovna, kterou můžeme stáhnout z webových stránek `http://am.vsb.cz/milota` v sekci download. ASU 10 obsahuje dvě procedury - intdraw a intraw3d, které vracejí na výstup Maplu 2D nebo 3D znázornění aproximace rovinných obrazců nebo těles v závislosti na pořadí integrace. Vstupem mohou být pouze proměnné $x, y, z, r, \theta, \phi, \psi$., přitom procedury automaticky rozpoznají, zda je těleso nebo rovinný obrazec zadán kartézskými, polárními, válcovými nebo sférickými souřadnicemi.

- intdraw(u=u1..u2,v=v1..v2,grid=[5,5],spces=[0.1,0.1]), kde u1,u2,v1,v2 jsou integrační meze, grid nastavuje počet objektů, jimiž je těleso aproximováno a spaces nastavuje velikost mezer mezi objekty, jimiž je těleso aproximováno

- intraw3d(u=u1..u2,v=v1..v2,w1..w2,grid=[5,5,5],spces=[0.1,0.1,0.1]), kde u1, u2, v1, v2, w1, w2 jsou integrační meze, grid nastavuje počet objektů, jimiž je těleso aproximováno a spaces nastavuje velikost mezer mezi objekty, jimiž je těleso aproximováno

Nyní si ukážeme použití ASU 10 na příkladě, kdy chceme vypočítat $\iint_M xy^2 \mathrm{d}x\mathrm{d}y$, kde množina $M = \{(x, y) \in \mathbb{R}^2, x^2 + 4y^2 \leq 1, x \geq 0\}$.

Naším cílem je vypočítat objem tělesa pomocí dvojného integrálu. Při definování pojmu Riemannův (dvojný) integrál vycházíme z tzv. integrálních součtů. Přitom víme, že dělíme-li integrační obor (množina, přes kterou integrujeme) na čím dál menší dílky, pak se integrální součty nezávisle na volbě reprezentantů čím dál více přibližují k pevné hodnotě I, kterou pak nazýváme dvojným Riemannovým integrálem. Na následujících obrázcích vidíme jednak těleso, jehož objem určujeme, jednak integrální součty pro různě hustá dělení integračního oboru.

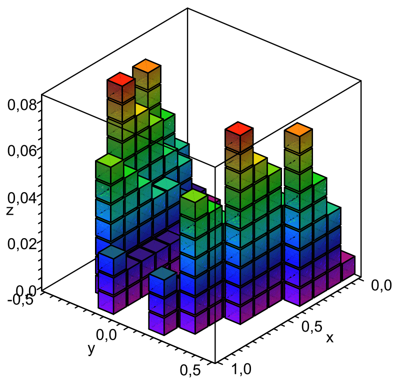
Aproximaci tělesa T, které je shora ohraničeno funkcí $f(x, y) = xy^2$ na množině M s mezemi

$$-\frac{1}{2} \leq y \leq \frac{1}{2}$$

$$0 \leq x \leq \sqrt{1 - 4y^2}$$

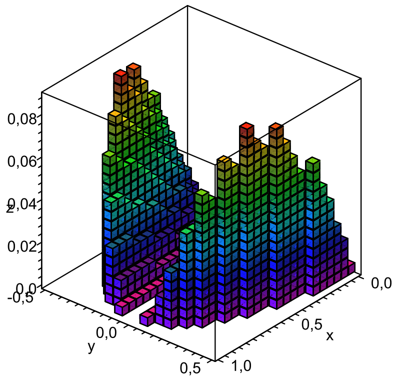
získáme následující posloupností příkazů. Nejdříve načteme balíček ASU 10 v prostředí Maplu a následně vykreslíme aproximované tělesa s různým počtem rozdělení.

```
> with(plots);
> read "asu10.txt";
> with(asu10);
> intdraw3d(y = -1/2..1/2, x = 0..sqrt(1-4*y^2), z = 0..x*y^2, grid =
[10, 10, 10], spaces = [.1, .1, .1], axes = boxed, labels = [x, y, z],
style = surfacewireframe, lightmodel = light2, glossiness = .5, shading
= zhue, orientation = [40, 60], scaling = unconstrained, title = "");
```



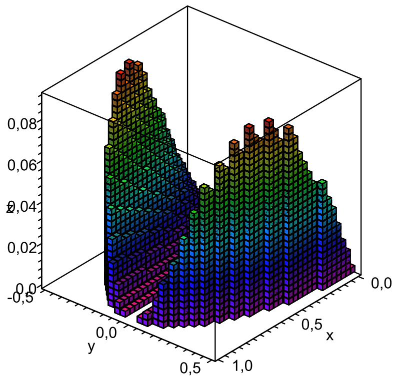
Obrázek 7: Graf vygenerovaný pomocí ASU při rozdělení [10,10,10]

```
>  intdraw3d(y = -1/2..1/2, x = 0..sqrt(1-4*y^2), z = 0..x*y^2, grid =
[20, 20, 20], spaces = [.1, .1, .1], axes = boxed, labels = [x, y, z],
style = surfacewireframe, lightmodel = light2, glossiness = .5, shading
= zhue, orientation = [40, 60], scaling = unconstrained, title = "");
```



Obrázek 8: Graf vygenerovaný pomocí ASU při rozdělení [20,20,20]

```
>  intdraw3d(y = -1/2..1/2, x = 0..sqrt(1-4*y^2), z = 0..x*y^2, grid =
[30, 30, 30], spaces = [.1, .1, .1], axes = boxed, labels = [x, y, z],
style = surfacewireframe, lightmodel = light2, glossiness = .5, shading
= zhue, orientation = [40, 60], scaling = unconstrained, title = "");
```



Obrázek 9: Graf vygenerovaný pomocí ASU při rozdělení [30,30,30]

Na obrázcích vidíme jak těleso, jehož objem určujeme, tak integrální součty pro různě hustá dělení integračního oboru.

3 Calcplot

K vytvoření grafických objektů pomocí systému Maple se nabízí použití balíčku "The Calcplot package for Maple", který vytvořil Timothy A. Mudroch. Tento balíček vznikl již v roce 1994 pro verzi Maple 5. Postupem času byl upravován a vznikla také počeštěná verze Calcrp5, která obsahuje stejné procedury s českou nápovědou. Balíček Calcplot, respektive Calcrp5 můžeme stáhnout z webových stránek `http://am.vsb.cz/milota` v sekci download, kde se taky nachází originální nápověda. V balíčku Calcplot nalezneme procedury, které vracejí na výstup Maplu 2D nebo 3D grafy. V každé proceduře jsou nastaveny základní atributy grafů, které jdou ovšem nastavit dle vlastních požadavků, jedná se například o styl, osvětlení, stínování a atd..
Základní hodnoty nastavené Calcplot jsou :

- axes=normal, scaling=constrained pro 2D graph
- axes=boxed, scaling=constrained, style=patch pro 3D graph.

3.1 Rotující křivka

Tato procedura vrací 3D graf. Navíc je možno použít jakákoliv nastavení plot3d a různé grafy mohou být kombinovány pomocí funkce display3D z balíčku plots. Je důležité si uvědomit, že pro x-ovou osu odpovídá $y = 0$ a pro y-ovou osu odpovídá $x = 0$.
Syntaxe :

- rotxplot(f, x=a..b, y=c, options)
  Tato procedura vykresluje povrch tělesa vzniklého rotací křivky okolo uživatelem specifikované osy. Ta je v tomhle případě rovnoběžná s x-ovou osou. Druhý argument určuje definiční obor nezávislé proměnné x a třetí argument osu rotace.
- rotyplot(f, x=a..b, x=c, options)
  Tato procedura vykresluje povrch tělesa vzniklého rotací křivky okolo uživatelem specifikované osy. Ta je v tomhle případě rovnoběžná s y-ovou osou. Syntaxe je identická jako rotxplot, kromě třetího argumentu, který udává rovnoběžnost s x nebo y osou a osou rotace.

3.2 2D grafy

Procedury vracející 2D graf. Je možné použít jakákoliv nastavení plot a různé grafy mohou být kombinovány pomocí funkce display z balíčku plots.
Syntaxe :

- dydxplot(y=f(x)..g(x), x=a..b, options)
  Tato procedura vykresluje obsah plochy v $\mathbb{R}^2$ mezi dvěma funkcemi $y = f(x)$ a $y = g(x)$ pro $a \leq x \leq b$. Doplňkem této procedury je vertikální šrafování na daném obsahu. Analogicky k této proceduře existuje i dxdyplot(x=h(y)..k(y), x=c..d, options) v $\mathbb{R}^2$ mezi dvěma funkcemi $x = h(y)$ a $x = k(y)$ pro $c \leq y \leq d$. Syntaxe je úplně stejná, jenom role x a y jsou obrácené a šrafování je v tomhle případě horizontální.

- drdtplot(r=f(θ)..g(θ), θ=a..b, options)
  Tato procedura vykresluje obsah plochy v $\mathbb{R}^2$ při použití mezí transformovaných do polárních souřadnic, meze jsou $r = f(\theta)$ a $r = g(\theta)$ pro $a \leq \theta \leq b$. Doplňkem této procedury je poloměrové šrafování na daném obsahu. Analogicky k této proceduře existuje i dtdrplot(θ=h(r)..k(r), r=c..d, options) v $\mathbb{R}^2$ s mezemi $\theta = h(r)$ a $\theta = k(r)$ pro $c \leq r \leq d$. Syntaxe je úplně stejná, jenom role r a t jsou obrácené.

3.3 3D grafy funkcí dvou proměnných

Procedury vykreslující grafy funkcí dvou proměnných. Každá z těchto procedur vrací 3D graf. Mohou být použity jakákoliv nastavení plot3d a různé grafy mohou být kombinovány pomocí funkce display3D z balíčku plots.
Syntaxe :

- yxgraphplot(q(x, y), y=f(x)..g(x), x=a..b, options)
  Tato procedura vykresluje funkci z=q(x, y) v $\mathbb{R}^3$ s mezemi $y = f(x)$ a $y = g(x)$ pro $a \leq x \leq b$. Analogicky k této proceduře existuje i yxgraphplot(q(x, y), x=h(y)..k(y), y=c..d, options) v $\mathbb{R}^3$ s mezemi $x = h(y)$ a $x = k(y)$ pro $c \leq y \leq d$. Syntaxe je úplně stejná, jenom role x a y jsou obrácené. Syntaxe obou procedur odráží podobnost ke dydxplot a dxdyplot, jenom je důležité si uvědomit, že je Maple vykresluje pomocí plot3d balíčku. Tato procedura dává stejný výsledek jako plot3d a dokonce i zápis je podobný.

- rtgraphplot(q(r, θ), r=f(θ)..g(θ), θ=a..b, options)
  Tato procedura vykresluje funkci z=q(r, θ) v $\mathbb{R}^3$ při použití funkce a mezí transformovaných do polárních souřadnic, meze jsou $r = f(\theta)$ a $r = g(\theta)$ pro $a \leq \theta \leq b$. Analogicky k této proceduře existuje i trgraphplot(Q(r, θ), θ=h(r)..k(r), r=c..d, options) v $\mathbb{R}^2$ s mezemi $\theta = h(r)$ a $\theta = k(r)$ pro $c \leq r \leq d$. Syntaxe je úplně stejná, jenom role r a t jsou obrácené. Syntaxe obou procedur odráží podobnost ke drdtplot a dtdrplot.

3.4 Zobrazení těles v prostoru

Tato kolekce obsahuje 18 různých procedur vykreslujících grafy, šest pro kartézské souřadnice (x, y, z), šest pro válcové souřadnice (r, t, z) a šest pro sférické souřadnice (ρ, ϕ, ψ). Každá z těchto procedur vrací plot3d graf. Je možné použít jakákoliv nastavení plot3d a různé grafy mohou být kombinovány pomocí funkce display3d z balíčku plots.
Syntaxe :

- dzdydxplot(z=f(x,y)..g(x,y), y=h(x)..k(x), x=a..b, options)
  Tato procedura vykresluje těleso v $\mathbb{R}^3$,které je popsáno pomocí kartézských souřadnic a je zdola ohraničeno funkcí $z = f(x, y)$ a shora funkcí $z = g(x, y)$, které leží v množině $M = \{(x, y) \in \mathbb{R}^2 : h(x) \leq y \leq k(x); a \leq x \leq b\}$. Analogicky k této proceduře existuje i dzdxdyplot, dydzdxplot, dydzdxplot, dxdydzplot, dxdzdyplot. Syntaxe je úplně stejná, jenom role x, y a z jsou přehozené.

- dzdrdtplot(z=f(r,θ)..g(r,θ), r=h(θ)..k(θ), θ=a..b, options)
  Tato procedura vykresluje těleso v $\mathbb{R}^3$,které je popsáno pomocí válcových souřadnic a je zdola ohraničeno funkcí $z = f(r, \theta)$ a shora funkcí $z = g(r, \theta)$, které leží v množině $M = \{(r, \theta) \in \mathbb{R}^2 : h(\theta) \leq r \leq k(\theta); a \leq \theta \leq b\}$. Analogicky k této proceduře existuje i dzdtdrplot, drdzdtplot, drdtdzplot, dtdrdzplot, dtdzdrplot. Syntaxe je úplně stejná, jenom role z, r a θ jsou přehozené.

- dpdtdphiplot(ρ=f(θ,ϕ)..g(θ,ϕ), θ=h(ϕ)..k(ϕ), ϕ=a..b, options)
  Tato procedura vykresluje těleso v $\mathbb{R}^3$,které je popsáno pomocí sférických souřadnic a je zdola ohraničeno funkcí $\rho = f(\theta, \phi)$ a shora funkcí $\rho = f(\theta, \phi)$, které leží v množině $M = \{(\theta, \phi) \in \mathbb{R}^2 : h(\phi) \leq \theta \leq k(\phi); a \leq \phi \leq b\}$. Analogicky k této proceduře existuje i dpdphidtplot, dtdpdphiplot, dtdphidpplot, dphidtdpplot, dphidpdtplot. Syntaxe je úplně stejná, jenom role ρ, θ a ϕ jsou přehozené.

3.5 Ukázky příkladů

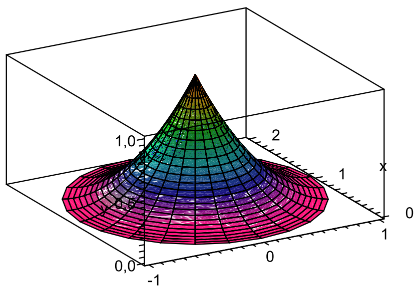
Balíček Calcplot, respektive jeho počeštěnou verzi Calcrp5 načteme v prostředí Maplu příkazy

```
>  read "calcplot.txt";
>  read "calcrp5.txt";
```

Nyní si ukážeme použití Calcplotu na pár příkladech. Je dobré si je vyzkoušet jak k naučení syntaxe, tak k otestování, že balíček funguje a instalace(načtení) proběhla úspěšně.

K vykreslení povrchu tělesa vzniklého rotací křivky okolo osy rovnoběžné s y-ovou osou vyzkoušejte následující :

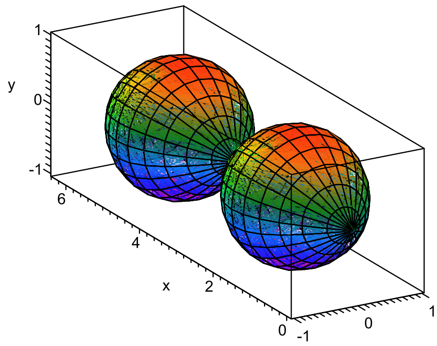
```
>  rotyplot(x^2, x = 0..1, x = 1, style = surfacewireframe, axes = boxed,
lightmodel = light2, glossiness = .5, shading = zhue, orientation = [240,
70], scaling = constrained);
```



Obrázek 10: Graf vygenerovaný pomocí funkce rotyplot

Nyní ukázka k vykreslení povrch tělesa vzniklého rotací křivky okolo osy rovnoběžné s x-ovou osou :

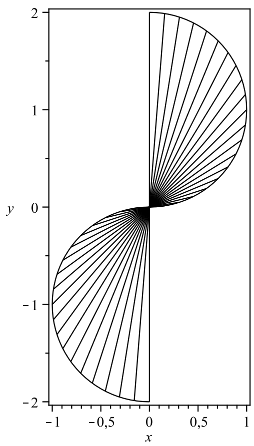
```
> rotxplot(sin(x), x = 0..2*Pi, y = 0, style = surfacewireframe, lightmodel
= light2, glossiness = .5, shading = zhue, orientation = [240, 70], scaling
= constrained);
```



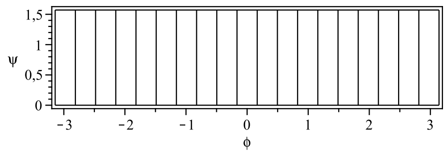
Obrázek 11: Graf vygenerovaný pomocí funkce rotxplot

Příklady v následující skupině zobrazují 2D grafy zadány pomocí kartézských nebo polárních souřadnic.

```
> drdtplot(r = -2*sin(theta)..2*sin(theta), theta = 0..(1/2)*Pi, axes
= boxed, labels = [x, y], scaling = constrained);
> dydxplot(y = 0 .. (1/2)*Pi, x = -Pi .. Pi, axes = boxed, labels = [phi,
psi], color = black, scaling = constrained);
```



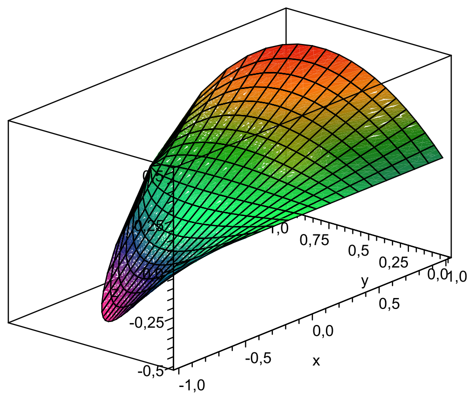
Obrázek 12: Graf vygenerovaný pomocí funkce drdtplot



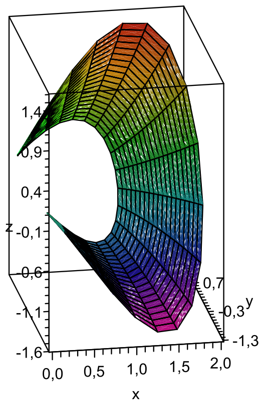
Obrázek 13: Graf vygenerovaný pomocí funkce dydxplot

Následující příklady vyobrazí funkci na požadované množině.

```
>  xygraphplot(x*y, x = -sqrt(1-y^2)..sqrt(1-y^2), y = 0..1, style = surface
wireframe, axes = boxed, lightmodel = light2, glossiness = .5, shading
= zhue, orientation = [220, 70], scaling = constrained);
>  rtgraphplot(r^2*sin(theta)*cos(theta), r = 1 .. 1+cos(theta), theta
= -(1/2)*Pi .. (1/2)*Pi, style = surfacewireframe, axes = boxed, lightmodel
= light2, glossiness = .5, shading = zhue, orientation = [260, 70], scaling
= constrained);
```



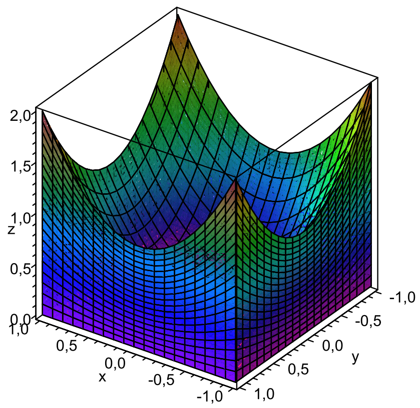
Obrázek 14: Graf vygenerovaný pomocí funkce xygraphplot



Obrázek 15: Graf vygenerovaný pomocí funkce rtgraphplot

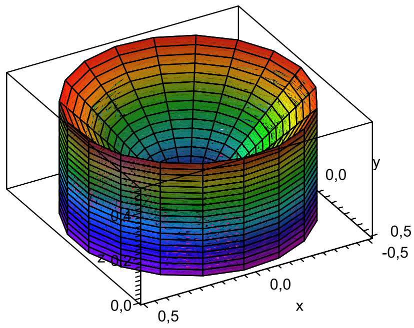
Nakonec si ukážeme jak vykreslit integrační obor pro trojný integrál v kartézských, válcových a sférických souřadnicích. Při transformaci do sférických souřadnic je nutné použít substituci ve tvaru $x = \rho\cos(\phi)\sin(\psi)$, $y = \rho\sin(\phi)\sin(\psi)$, $z = \rho\cos(\psi)$, což je standardní tvar pro systém Maple.

```
>  dzdydxplot(z = 0..x^2+y^2, y = -1..1, x = -1..1, axes = boxed, labels
= [x, y, z], style = surfacewireframe, lightmodel = light2, glossiness
= .5, shading = zhue, orientation = [125, 60], scaling = unconstrained);
```



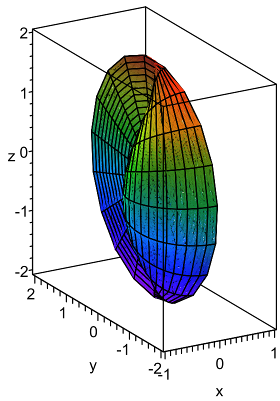
Obrázek 16: Graf vygenerovaný pomocí funkce dzdydxplot

```
>  dzdrdtplot(z = 0..r, r = 0..1/2, theta = -Pi..Pi, axes = boxed, labels
= [x, y, z], style = surfacewireframe, lightmodel = light2, glossiness
= .5, shading = zhue, orientation = [60, 60], scaling = constrained);
```



Obrázek 17: Graf vygenerovaný pomocí funkce dzdrdtplot

```
>  dpdtdphiplot(rho = 0..2*sin(theta), theta = 0..Pi/2, phi = -Pi..Pi,
axes = boxed, labels = [x, y, z], style = surfacewireframe, lightmodel
= light2, glossiness = .5, shading = zhue, orientation = [240, 70],   scaling
= constrained);
```



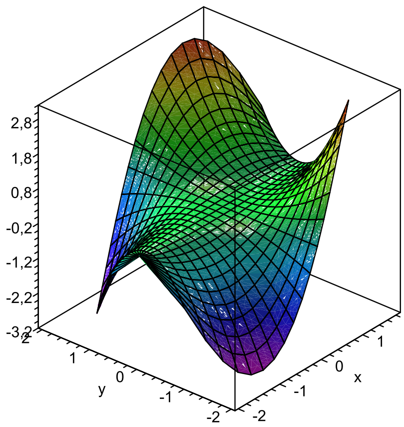
Obrázek 18: Graf vygenerovaný pomocí funkce dpdtdphiplot

3.6 Grafika ke kapitole transformace do polárních souřadnic

Mějme zadán příklad : Vypočtěte $\iint_M xy^2 \mathrm{d}x\mathrm{d}y$, kde množina $M = \{(x,y) \in \mathbb{R}^2, x^2 + y^2 \leq 4\}$.

Je vhodné si vykreslit funkci $f(x,y) = xy^2$ na množině M. K tomuto účelu můžeme použít mapleovskou funkci plot3d. Ta je zadána níže uvedeným příkazem.

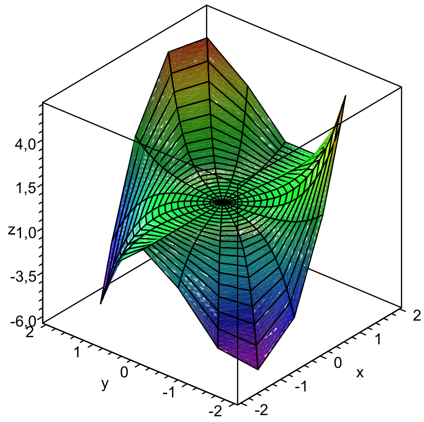
```
> plot3d(x*y^2, x = -2..2, y = -sqrt(4-x^2)..sqrt(4-x^2), axes = BOXED,
labels = [x, y, z], style = surfacewireframe, lightmodel = light2, glossiness
= .5, shading = zhue, orientation = [220, 60], scaling = unconstrained);
```



Obrázek 19: Vykreslení funkce použitím plot3d

Druhou možností jak požadovanou funkci vykreslit je použití Calcplot funkce rtgraphplot, v níž je funkce zadána po transformaci do polárních souřadnic a zobrazena v kartézské soustavě souřadnic.

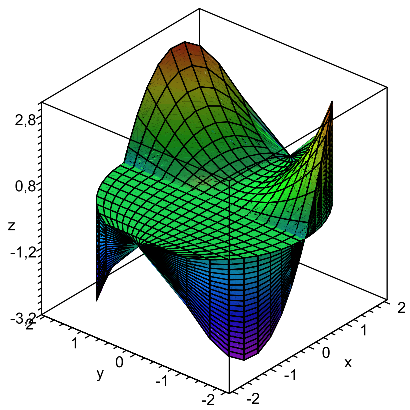
```
> rtgraphplot((r*cos(theta)*(r^2))*sin(theta)^2, r = 0..2, theta = -Pi..Pi,
axes = BOXED, labels = [x, y, z], style = surfacewireframe, lightmodel
= light2, glossiness = .5, shading = zhue, orientation = [220, 60], scaling
= unconstrained);
```



Obrázek 20: Vykreslení funkce použitím rtgraphplot

Jestliže uvažujeme dvojný integrál jako objem tělesa, které je shora ohraničeno funkcí $f(x,y) = x * y^2$ na množině M, můžeme si toto těleso vykreslit pomocí Calcplot funkce dzdydxplot. Funkce je zadána v kartézských souřadnicích a zobrazena v kartézské soustavě souřadnic.

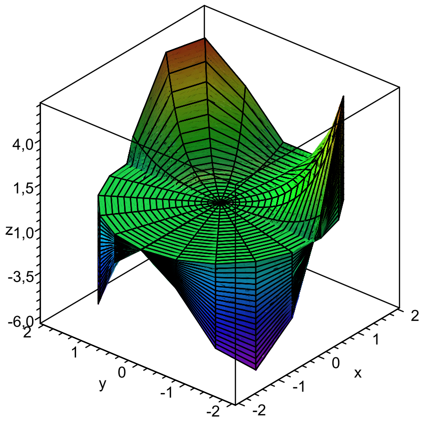
```
> dzdydxplot(z = 0..x*y^2, y = -sqrt(4-x^2)..sqrt(4-x^2), x = -2..2,
axes = BOXED, labels = [x, y, z], style = surfacewireframe, lightmodel
= light2, glossiness = .5, shading = zhue, orientation = [220, 60], scaling
= unconstrained);
```



Obrázek 21: Vykreslení tělesa použitím dzdydxplot

Druhou možností vykreslení tělesa je použití Calcplot funkce dzdrdtplot, v níž je těleso zadáno po transformaci do polárních souřadnic a zobrazeno v kartézské soustavě souřadnic.

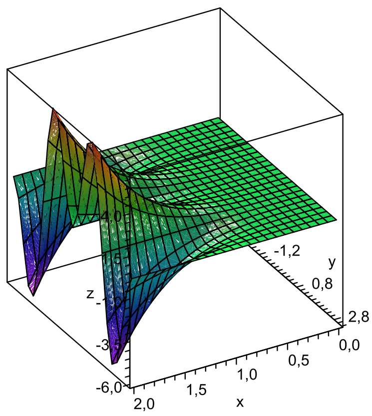
```
> dzdrdtplot(z = 0..(r*cos(theta)*(r^2))*sin(theta)^2, r = 0..2, theta
= -Pi..Pi, axes = BOXED, labels = [x, y, z], style = surfacewireframe,
lightmodel = light2, glossiness = .5, shading = zhue, orientation = [220,
60], scaling = unconstrained);
```



Obrázek 22: Vykreslení tělesa použitím dzdrdtplot

K vykreslení funkce, která vznikne transformací do polárních souřadnic můžeme použít mapleovskou funkci plot3d.

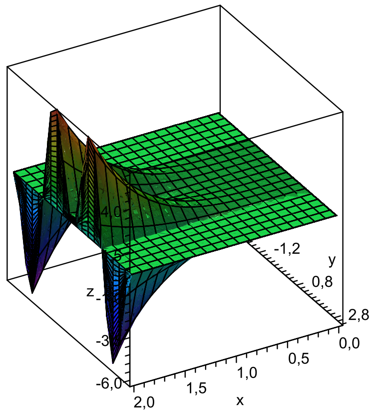
```
> plot3d((r*cos(theta)*(r^2))*sin(theta)^2, r = 0..2, theta = -Pi..Pi,
axes = BOXED, labels = [x, y, z], style = surfacewireframe, lightmodel
= light2, glossiness = .5, shading = zhue, orientation = [60, 60], scaling
= unconstrained);
```



Obrázek 23: Vykreslení funkce po transformaci použitím plot3d

Opět, chápeme-li dvojný integrál jako objem tělesa, můžeme použít Calcplot funkci dzdydxplot. Funkce je zadána po transformaci v kartézských souřadnicích a zobrazena v kartézské soustavě souřadnic.

```
> dzdydxplot(z = 0..(x*cos(y)*(x^2))*sin(y)^2, y = -Pi..Pi, x = 0..2,
axes = BOXED, labels = [x, y, z], style = surfacewireframe, lightmodel
= light2, glossiness = .5, shading = zhue, orientation = [60, 60], scaling
= unconstrained);
```



Obrázek 24: Vykreslení tělesa po transformaci použitím dzdydxplot

4 JavaView

JavaView je software pro zobrazení a manipulaci interaktivních 3D objektů vyexportovaných pomocí mapleovské knihovny JavaViewLib, která spojuje programy Maple a JavaView při zachování všech výhod obou programů. Pomocí JavaView můžeme jednoduše vytvářet a upravovat 3D modely, které mohou být následně umístěny v HTML stránkách. Tyto modely si uchovávají veškerou interaktivitu jako je přibližování, rotace nebo posunutí. Modely se do HTML stránek vkládají jako applet, protože JavaView je program napsaný v programovacím jazyce Java, který využívá všech jeho výhod jako jsou například nezávislost na platformě, operačním systému nebo prohlížeči.

4.1 Stažení a instalace JavaView

Stáhnutí a instalace programu JavaView není nikterak složité. Z webové adresy `http://www.javaview.de/download/index.html` máme na výběr různé verze JavaView pro různé účely. Po stažení instalační verze `javaviewWin_setup.exe` nás průvodce provede instalací. Nám dostačující balík s knihovnou pro integraci do programů Maple `jv_maple.zip` stáhneme a následně rozbalíme do instalačního adresáře Maplu, například `C:\Program Files\Maple 12`. Archív může být rozbalen a umístěn i jinde, autoři ho však doporučují mít ve složce, kde je Maple nainstalován, pravděpodobně z důvodu bezproblémového načtení přímo v Maplu.

4.2 Ovládání JavaView appletů

Pomocí stisknutí a pohybem myší nebo pomocí klávesových zkratek je možné s objektem v okně appletu pohybovat, otáčet, přibližovat atd.. Výchozí mód je nastaven na otáčení. Pro otáčení stiskneme levé tlačítko myši a pohybujeme požadovaným směrem. Jestliže chceme změnit mód, vybereme jeho klávesovou zkratku a postupujeme stejně. Pro použití základních operací lze použít pravé tlačítko myši v okně appletu a zvolit ji v nabídce. Klávesové zkratky operací jsou uvedeny v tabulce 1 : Ovládání JavaView appletů.

4.3 Export grafů z Maplu

JavaView umožňuje import a export 3D objektů různých formátů jako například JVX, VRML, OBJ a formátu souboru s grafikou programů Maple nebo Mathematica. JavaView uchovává všechny možnosti úpravy grafických modelů. JavaViewLib umožňuje export interaktivní 3D grafiky z Maplu do formátu MPL (interní datový formát Maplu) a JVX (oficiální formát programu JavaView). Navíc se dají grafické modely exportovat přímo do HTML kódu, který je automaticky vygenerován včetně nastavení a vložení appletu.

Jestliže máme nainstalován Maple a JavaView rozbaleno v jeho adresáři dle předchozích pokynů, můžeme začít používat funkce této knihovny. Prvním krokem je nastavení cesty k adresáři JavaViewLib a následně samotné načtení, to provedeme příkazy :

```
> libname := "C:\Program Files\Maple 12\JavaViewLib\";
> libname;
> with(JavaViewLib);
```

Knihovna JavaViewLib obsahuje mnoho funkcí, po zadání příkazu k načtení knihovny se nám dostupné funkcí vypíšou. Jsou to *exportHTM, exportHTMLite, exportJVX, exportMPL, exportValidate, genTag, genTagLite, getBrowser, getInfo, getInfoState, getOS, import, importJVX, importMPL, runApplet, runAppletLite, runJavaView, runMarkupTree, set, setEnabledValidate, setWorkingPath, viewGallery*.

K exportu mapleovských grafů jsou nejpotřebnější funkce *exportHTM, exportHTMLite, exportJVX, exportMPL*, které si trochu popíšeme.

- exportMPL - zadávaný příkazem exportMPL(graf, ”soubor”), kde graf je parametr, který udává co má být exportováno a parametr soubor značí název exportovaného souboru spolu s příponou a popřípadě cestu kam to má být exportováno. Exportuje graf do souboru formátu MPL, což je textový soubor reprezentující graf.

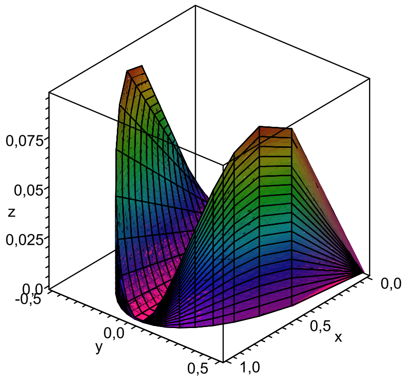
- export JVX - zadávaný příkazem exportJVX(graf, ”soubor”). Exportuje graf do souboru formátu JVX, což je XML reprezentace grafu.

- exportHTM - zadávaný příkazem exportHTM(graf, ”soubor”). Vytváří jednoduchý HTML dokument obsahující graf. Pokud určíme příponu souboru .mpl, exportuje se graf do formátu MPL. V případě neuvedení přípony se graf exportuje v podobě kódu přímo v appletu mezi tagy `<APPLET></APPLET>`.

- exportHTMLite - zadávaný příkazem exportHTMLite(graf, ”soubor”). Oproti funkci exportHTM se odlišuje v tom, že se v HTML stránce použije odlehčená verze appletu JavaView a to jvLite.jar.

Příklady použití :

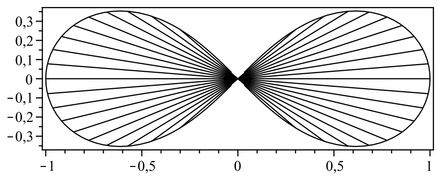
```
> priklad := dzdxdyplot(z = 0..x*y^2, x = 0..sqrt(1-4*y^2), y = -1/2
..1/2, axes = boxed, labels = [x, y, z], style = surfacewireframe, lightmodel
= light2, glossiness = .5, shading = zhue, orientation = [40, 60], scaling
= unconstrained);
> exportMPL(priklad, "cesta\MPLexport\ priklad.mpl ");
```



Obrázek 25: Ukázka grafu vyexportovaného pomocí JavaView

Dle mého názoru je nejednodušší upravit grafy v Maplu pomocí nastavení, které jsme si popsali v kapitole 2 na straně 7, a následně exportovat do formátu MPL, který zachovává většinu tohoto nastavení. Někdy se ovšem stane, že JavaView všechny nastavení nevyexportuje, potom můžeme soubory ještě upravit programem JavaView. Týká se to například nastavení průhlednosti, Transparency, u které se mi nepodařilo dodatečně, za pomoci JavaView, dosáhnout stejného vizuálního efektu jako při použití Maplu, takže jsem ji nakonec vynechal. Dále se někdy při exportování přihodilo, že se z nevysvětlitelného důvodu zbarvili některé osy do barvy modré, i když to tak nebylo nastaveno. Za zmínku stojí i to, že ne všechny exporty grafů pomocí JavaView musí proběhnout úspěšně. V jednom případě jsem se snažil zapsat požadovaný výstup pomocí všech dostupných funkcí, ale ani přesto se mi 2D graf nevyexportoval. Týká se to například příkazu :

```
> drdtplot(r = -sqrt(cos(2*theta))..sqrt(cos(2*theta)), theta = -(1/4)
*Pi..(1/4)*Pi, labels = ["", ""], axes = BOXED, title = "", color = black,
scaling = unconstrained);
```



Obrázek 26: Ukázka grafu, který nelze vyexportovat pomocí JavaView

Nakonec jsem musel tento interaktivní obrázek v rovině nahradit klasickým obrázkem, což naštěstí v tomhle případě nevadí, horší by to bylo s obrázkem v 3d prostoru. V každém případě, kapitolou JavaView a jejím nastavením se podrobněji zabývá má kolegyně Michaela Vondrová ve své bakalářské práci a taktéž o knihovně JavaView bylo napsáno již hodně publikací.

Transformace	
o	Rotace objektu
s	Změna měřítka
t	Posun objektu
c	Zaměření pohledu na střed objektu
f	Zaměření pohledu na střed objektu při zachování měřítka
r	Reset původního nastavení objektu
x	Přiblížení obdélníkového výřezu
Ovládání animace	
w	Spuštění animace naposledy použité transformace
q	Zastavení animace
Ctrl-a	Zobrazení panelu ovládání animace
Ovládání modelace objektu	
a	Přidání vrcholu objektu
e	Vytvoření nového prvku nebo polygonu přidáním nových vrcholů
d	Vytvoření kopie veškeré zobrazené geometrie
i	Posun výchozího bodu
m	Označení vrcholů v obdélníkovém výřezu
n	Označení plošek (prvků) v obdélníkovém výřezu
p	Uchopení a přesun vrcholu
u	Odznačení vrcholů v obdélníkovém výřezu
Shift-u	Odznačení plošek (prvků) v obdélníkovém výřezu
Ovládání výběru	
Ctrl-levé tlačítko myši	Výběr objektu
Ctrl-tab	Přepínání mezi jednotlivými objekty
Mazání	
Ctrl-Shift-y	Vymazání všech objektů
Delete-levé tlačítko myši	Vymazání objektu
Zobrazení ovládacích panelů	
Fl	Zobrazení okna ovládacího panelu
F2	Uzavření okna ovládacího panelu
Ctrl-a nebo F4	Zobrazení okna ovládání animace
Ctrl-c	Zobrazení informačního okna kamery
Ctrl-d nebo F3	Zobrazení debug okna
Ctrl-i	Zobrazení okna s informacemi o zobrazeném objektu
Ctrl-1	Zobrazení okna s nastavením osvětlení
Ctrl-m	Zobrazení okna s nastavením zobrazení objektu
Ctrl-t	Zobrazení panelu s nastavením textury zobrazeného objektu
Klávesové zkratky MENU	
Ctrl-n	Otevře okno s novým objektem
Ctrl-Shift-n	Načte JVD soubor se stejným jménem jako naposledy otevřený soubor
Ctrl-o	Otevře okno s importem souborů
Ctrl-Shift-o	To samé jako Ctrl+O, ale nahrává taky JVD soubor jestliže je dostupný
Ctrl-s	Otevře okno pro uložení objektu do různých formátů
Nouzové ukončení JavaView	
Ctrl-Shift-x	Ukončí neprodleně JavaView

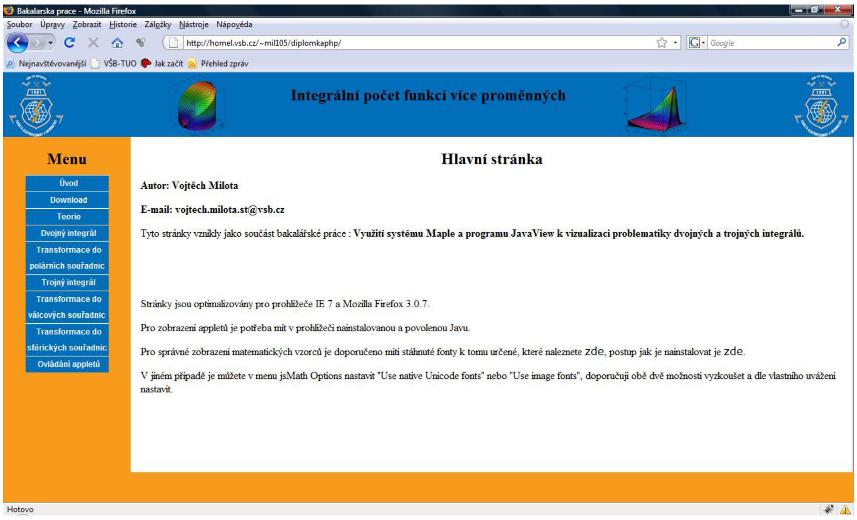
Tabulka 1: Ovládání JavaView appletů

5 HTML stránky

Jak jsem se již zmínil v úvodu, cílem této bakalářské práce je prezentace interaktivních grafických objektů na webu. K vytvoření těchto stránek můžeme použít funkce JavaView, avšak webová galerie není moc pohledná. Proto jsem navrhl a následně vytvořil stánky vlastní. Základní znalosti web designu a programování stránek pomocí technologií jako je CSS, XHTML a PHP, mi pomohli vytvořit web pro požadované účely. Ve finální fázi byla vytvořena webová galerie s třiceti příklady rozdělených do pěti částí a navíc byla vytvořena část obsahující veškerou potřebnou teorii ke spočítání všech těchto příkladů. Webové stránky obsahují :

- Teorie - Obsahuje teorii rozdělenou dle použití k jednotlivým kapitolám.
- Dvojný integrál - Obsahuje 8 příkladů výpočtu dvojného integrálu pomocí fubiniovy věty, kde každý příklad je doplněn interaktivními obrázky množiny přes kterou se integruje a také integrovanou funkcí.
- Transformace do polárních souřadnic - Obsahuje 6 příkladů výpočtu dvojného integrálu pomocí transformace do polárních souřadnic, kde každý příklad je doplněn interaktivními obrázky množiny přes kterou se integruje a integrované funkce před i po transformaci.
- Trojný integrál - Obsahuje 6 příkladů výpočtu trojného integrálu pomocí fubiniovy věty, kde každý příklad je doplněn interaktivními obrázky množiny přes kterou se integruje.
- Transformace do válcových souřadnic - Obsahuje 5 příkladů výpočtu trojného integrálu pomocí transformace do válcových souřadnic, kde každý příklad je doplněn interaktivními obrázky množiny přes kterou se integruje a průmětem do rovin před i po transformaci.
- Transformace do sférických souřadnic - Obsahuje 5 příkladů výpočtu trojného integrálu pomocí transformace do sférických souřadnic, kde každý příklad je doplněn interaktivními obrázky množiny přes kterou se integruje a průmětem do rovin před i po transformaci.

Na obrázku je úvodní stránka webové galerie.



Obrázek 27: Náhled úvodní stránky webové galerie

5.1 Programování stránek a PHP

Stránky můžeme programovat v HTML editoru PSPad, jenž je volně šiřitelný software, který je pro tuto práci přímo předurčen a obsahuje mnoho užitečných funkcí, které šetří čas. Jsou jimi například předdefinované šablony pro psaní PHP, XHTML nebo JavaScript dokumentů nebo projektový management. Více o tomto programu naleznete na jeho webových stránkách `http://www.pspad.com/cz/`. V první fázi jsem programoval stránky způsobem, kdy byla vytvořena kostra stránky, to znamená vytvořeny jednotlivé layouty jako je záhlaví, zápatí a menu stránky pomocí kaskádových stylů CSS a jednotlivé stránky byly vyplňovány potřebným textem. Tato metoda se však stala v momentu, kdy jsou stránky již naplněny špatná, protože jestliže chceme změnit byť jedinou věc v menu, musíme předělávat všechny vytvořené stránky. Ve chvíli, kdy si uvědomíme, že toho musíme více předělat, vzniká tak zdlouhavá otrocká práce. Této situaci lze předejít využitím základních funkcí PHP. Pomocí PHP si můžeme rozdělit jednotlivé části stránky na funkce a v jednotlivých stránkách potom tyto funkci volat. Definitivně tak oddělíme jak kostru stránek od jejího obsahu, tak i jednotlivé obsahové části stránek. Při změně položky v layoutu menu tak stačí změnit potřebnou PHP funkci v jednom souboru. Volání požadované funkce zůstává stejné. Příklad použití na mých stránkách je uveden na nadcházejících výpisech.

```
function footer_zpetvpred($adr1,$adr2)
{
?>
    <div id="footer">
        <div class="left"><a href="<?echo $adr1;?>">&lt;&lt; Zpět </a></div>
        <div class="right"><a href="<?echo $adr2;?>"> Další &gt;&gt;</a></div>
        <div class="clearer"></div>
    </div>

    <SCRIPT>
        jsMath.ConvertTeX();
        jsMath.Process(document);
    </SCRIPT>
  </body>
</html>
<?
}
```

Výpis 1: Příklad PHP funkce

Tato funkce uložena jako soubor function.php definuje zápatí stránky, které obsahuje posuvníky zpět a vpřed a obsahuje část scriptu jsMath, který překládá latexovský kód. Na dalším výpise je uvedeno jak tuto funkci načíst (je v ní taky uvedeno adresování stránek zpět a vpřed).

```
<?
  include "cesta/function.php";
  footer_zpetvpred('pr1_1.php','pr1_2.php');
?>
```

Výpis 2: Načtení PHP funkce v jednotlivých stránkách

5.2 Nastavení appletů

Po vygenerování souborů mpl, obsahující potřebné informace o modelu, pomocí JavaView se tyto modely zobrazují na stránkách prostřednictvím java appletů, kde je soubor mpl načten. K tomu aby byly tyto applety na stránkách funkční potřebujeme překopírovat na webový server adresář jars nacházející se ve složce JavaViewLib(viz strana 28), který obsahuje příslušné Java knihovny. Jsou to :

- javaview.jar - Java knihovna nezbytná pro chod appletů obsahující aplikaci Java View a Java třídy.
- jvx.jar - Java knihovna obsahující Java třídy pro programování složitějších algoritmů

- vgapp.jar - Java knihovna obsahující geometrické objekty k vytváření modelů pomocí JavaView
- jvLite.jar - Java knihovna obsahující základní Java třídy k vizualizaci modelů

Následně nahrajeme java applet do html kódu příkazem uvedeným na nadcházejícím výpisu.

```
<APPLET CODE='javaview.class' ARCHIVE='jars/javaview.jar,jars/jvx.jar' CODEBASE='.'
WIDTH='400' HEIGHT='400' ID='JVLAPPLET' ALT='JVL_–_MAPLE_Export'>
    <PARAM NAME='DepthCue' VALUE='hide'>
    <PARAM NAME='Background' VALUE='255_255_255'>
    <PARAM NAME='Border' VALUE='hide'>
    <PARAM NAME='Model' VALUE='mpl/2/fub/5/pr5_1.mpl'>
</APPLET>
```

Výpis 3: Načtení appletu v jednotlivých stránkách

Mezi párovými tagy `<APPLET></APPLET>` se nachází atributy, který tento applet nastavují. Základní atributy jsou :

- APPLET CODE - Název Java třídy spouštějící applet
- ARCHIVE - Nastavuje relativní cestu k adresáři jars a použitým java knihovnám
- CODEBASE - Nastavuje cestu od HTML stránky ke všem souborům použitým při zobrazení jako jsou knihovny JavaView a potřebné mpl soubory. V mém případě byly tyto složky umístěny ve složce se stránkami a proto je nastaveno CODEBASE='.', v případě, že by jsme měli soubory mpl, jars a stránky ve složkách na stejné úrovni bylo by nastavení CODEBASE='..'
- WIDTH, HEIGHT - Nastavuje šířku a výšku appletu

Další nastavení appletů týkající se zobrazovaného objektu je uvedeno v následující tabulce.

Param name	Value	Popis
AntiAlias	hide/show	Zjemňuje vykreslené hrany
Axes	hide/show	Zobrazuje souřadnicové osy
Background	255 255 255	Nastavuje barvu pozadí v RGB
BackgroundImage	URL nebo soubor	Možnost nahrání obrázku do pozadí appletu
BackgroundImageFit	center/fit	Pozice, respektive umístění obrázku v appletu
Border	hide/show	Nastavuje, zda má být zobrazen rámeček okolo appletu
BoundingBox	hide/show	Ohraničení kolem appletu
Copyright	hide/show	Zobrazení autorských práv JavaView
DepthCue	hide/show	Nastavuje stínování objektu
EdgeAura	hide/show	Zobrazí osvětlení kolem hran
ForegroundImage	URL nebo soubor	Možnost nahrání obrázku do popředí appletu
ForegroundImageFit	center/fit	Pozice, respektive umístění obrázku v appletu
Frame	hide/show	Zobrazuje souřadnicové osy v levém rohu appletu
Model	cesta k souboru	Nastavuje cestu k souboru s požadovaným objektem
Sorting	hide/show	Nastavuje malířův algoritmus pro zobrazení scény
ZBuffer	hide/show	Nastavuje z-buffer algoritmus pro zobrazení scény

Tabulka 2: Nastavení appletů

6 jsMath

Balíček jsMath poskytuje metody zobrazení matematiky v HTML stránkách, které fungují na prohlížečích pod Windows, Macintosh OS X a Linux. Překonává řadu nedostatků tradiční metody použití obrazů jako matematického výstupu. JsMath využívá k zobrazení matematiky nativní fonty, které jsou přizpůsobitelné k ostatním prvkům stránek, můžete tak bezproblémově měnit velikost textu ve vašem prohlížeči nebo stránky tisknout v plném rozlišení a nemusíte čekat na tucty obrázků, které by musely být stáhnuté. Tento balíček znatelně zjednodušuje autorům stránek vkládání matematiky, která tak nemusí být složitě stylizována formou obrázků, ale rozvržena jako jednoduchý text v odstavcích.

Ačkoliv jsMath pracuje nejlépe s nainstalovanými TeXovými fonty, má v záloze i obrázkové fonty (které jsou přizpůsobené k zvětšování nebo tištění ve vysokém rozlišení) nebo Unicode fonty když nejsou TeXové fonty dostupné. Na stránkách s použitým jsMath je ovládací panel v podobě malého plovoucího knoflíku, který nechává na uživateli, kterou nouzovou metodu použít nebo měnit další nastavení, například měřítko pro matematiku ve srovnání s okolním textem na stránce aj..

JsMath balík je založený na TeXovém výstupu matematiky při zachování co největší podobnosti s ním. Byl vytvořen k co nejjednodušší cestě jak dostat matematiku na web bez znalosti obtížného programování jako je například MathML, které navíc nezobrazuje na všech prohlížečích matematiku přesně.

6.1 Stažení a instalace jsMath

Používání jsMath k prezentaci matematiky v našich vlastních webových stránkách není těžké. Je zapotřebí stáhnout oba ”jsMath”a ”jsMath Image Fonts”balíčky z webové adresy `http://www.math.union.edu/~dpvc/jsMath/` v sekci Download jsMath. Pro instalaci jsMath na serveru stáhneme a rozbalíme oba dva archivy popsané výše a přesuneme vše co obsahují do složky nazvané jsMath kdekoliv na našem webovém serveru.

”jsMath Image Fonts”archiv obsahuje obrázkové soubory (v různé velikosti), které jsou používány v případě, kdy nejsou nainstalovány TeXové fonty. Obrázky jsou uložené v podadresářích složky ”font”, která by měla být umístěna ve složce jsMath na serveru. Obrázkové fonty zahrnují tisíce malých obrazových souborů (jeden pro každý znak ve všech velikostech) a ačkoliv jsou soubory dost malé, celkově zaberou hodně prostoru. Jestliže máme vše staženo a umístěno na serveru, můžeme si ověřit funkčnost jsMath na souboru index.html, který nalezneme ve složce test. Jestliže vidíme matematický text na webové stránce a zelený nápis je jsMath funkční a instalace proběhla v pořádku. V jiném případě musíme překontrolovat jednotlivé kroky instalace.

6.2 Sazba matematiky v jsMath

Matematiku do HTML stránky vkládáme používáním standardní TEXovské sazby. Matematika je v dokumentu označena užitím párových tagů SPAN nebo DIV třídou

CLASS=”math”. JsMath hledá tyto tagy a vysází TEX kód, který obsahují.

Tato metoda je možná, ale pro většinu nevyhovující, protože musíme psát tyto obklopující tagy kdykoliv chceme vypsat nějakou matematiku. Z tohoto důvodu jsMath obsahuje tex2math plugin, který skenuje dokument pro TEX a LATEX matematické omezovače

```
$...$, $$...$$, \(...\) a \[...\]
```

a nahrazuje tagy SPAN nebo DIV automaticky.

JsMath je taky vysoce přizpůsobitelný a má různé rozšíření a pluginy. Jejich nastavení není však pro úplného začátečníka lehké. Naštěstí má jsMath ”easy load”doplněk, kde můžeme nastavit hlavní detaily jsMath na jednom místě, a měl by stačit pro většinu autorů webových stránek. Jednoduše nahrajete jeden JavaScript soubor a zbytek nastavení je provedeno automaticky.

6.3 EasyLoad

Script ”load.js”nahrajeme příkazem ¡script src=”cesta-k-jsMath / easy / load.js»¡ / script¿. Je dobré si vytvořit vlastni ”load.js”s požadovaným nastavením (popřípadě více verzí s různým nastavením pro různé situace) a původní pro jistotu zálohovat. Jak vypadá hlavička dokumentu HTML je uvedeno na výpisu.

```
<!DOCTYPE html PUBLIC ”−//W3C//DTD␣XHTML␣1.0␣Strict//EN”
  ”http :// www.w3.org/TR/2000/REC−xhtml1−20000126/DTD/xhtml1−strict.dtd”>
<html xmlns=”http://www.w3.org/1999/xhtml” xml:lang=”cs” lang=”cs”>
<head><!−− ščřžýŠČŘŽÝ −−>
   < title >nějaký title</ title >
   <meta http−equiv=”content−type” content=”text/html;␣charset=windows−1250” />
   <meta name=”description” CONTENT=”nějaký␣název␣”>
   <meta name=”author” CONTENT=”nějaký␣autor”>
   <link rel=”stylesheet” type=”text/css” href=”cesta−k−stylum/styl.css” media=”screen”/>
   <script src=”cesta−k−jsMath/easy/load.js”></script>
</head>
```

Výpis 4: Načtení jsMath do HTML

Nyní přejděme k nastavení přímo ve scriptu ”load.js”. Ten obsahuje řadu funkcí s předvoleným nastavením, které se vetšinou upravuje pomocí hodnot 0 a 1, kde 0=vypnuto a 1=zapnuto nebo výrazy k tomu určenými, které si popíšeme níže.

- root
  Jestliže voláte v HTML script load.js a nemáte z jakéhokoliv důvodu celý adresář jsMath umístěn na stejném serveru, můžete ho načíst pomocí *root: ”http://stranky-kde-je-jsMath/cesta-k-jsMath”* .

- scale
  Určuje velikost matematiky k okolnímu textu, například *scale: 120* .

- autoload
  Prohledá HTML kód a zjišťuje zda je zapotřebí přeložit matematiku pomocí jsMath, defaultně nastaveno *autoload: 1* .
- processSlashParens
  Hledá a překládá v textu `\(...\)`, výchozí nastavení *processSlashParens: 1* .
- processSlashBrakets
  Hledá a překládá v textu `\[...\]`, výchozí nastavení *processSlashBrackets: 1* .
- processDoubleDollars
  Hledá a překládá v textu `$$...$$`, výchozí nastavení *processDoubleDollars: 1* .
- processSingleDollars
  Hledá a překládá v textu `$...$`, výchozí nastavení *processSingleDollars: 0* .
- processLaTeXenvironments
  Hledá a překládá v textu `\begin{xxx}...\end{xxx}`, výchozí nastavení *processLaTeXenvironments : 0* .
- fixEscapedDollars
  Překládá $ jako dolar $, výchozí nastavení *fixEscapedDollars: 0* .
- doubleDollarsAreInLine
  Překládá `$$...$$` jako matematiku zapsanou v jednom řádku, výchozí nastavení *doubleDollarsAreInLine: 0* .
- allowDisableTag
  Povoluje `ID="tex2math_off"` k vypnutí tex2math pluginu, výchozí nastavení *allowDisableTag: 1* .
- customDelimiters
  Dovoluje nadefinovat vlastní uvození matematiky, například *customDelimiters: ['[math]', '[/math]', '[display]', '[/display]']* k začátku a konci matematiky na řádku a zobrazení matematiky.
- safeHBoxes
- allowDoubleClicks
  Umožňuje po dvojkliku na matematiku zobrazit původní TEX zápis, *allowDoubleClicks: 1* .
- showFontWarnings
  Zobrazuje varování v případě, kdy koncový uživatel nemá nainstalovány TEX fonty, *showFontWarnings: 1* .
- method

Nastavuje metodu zobrazení matematiky, a to ”Process”, který zobrazí webovou stránku a postupně překládá TEX zápis nebo ”ProcessBeforeShowing”, který nejprve přeloží stránku a až následně ji zobrazí, načítání je však delší. Výchozí nastavení je *method: ”Process”* .

- loadFiles
  Načte další speciální doplňky jsMath, například *loadFiles: [”plugins/mimeTeX.js”, ”extensions/AMSsymbols.js”]* .
- loadFonts
  Načte dodatečně stáhnuté rozšířené fonty, například *loadFonts: [”cmmib10”]* .
- macros
  Umožňuje vytvářet makra.
- allowGlobal
  Překládá do mezipaměti následující stránky a tím urychluje zobrazení matematiky, přednastaveno na *allowGlobal: 1* .
- noImageFonts
  Vypíná načtení obrázků místo fontů, pokud je nemáte nahrané na serveru, základní nastavení *noImageFonts: 0* .

Po uvážení co a jak nastavit může náš script vypadat podobně jako je uvedeno v nadcházejícím výpisu.

```
root: "http :// stranky−kde−je−jsMath/cesta−k−jsMath",

scale: 125,

autoload: 1,

processSlashParens: 1,
processSlashBrackets: 1,
processDoubleDollars: 1,
processSingleDollars: 1,
processLaTeXenvironments: 1,
fixEscapedDollars: 1,
doubleDollarsAreInLine: 0,
allowDisableTag: 1,

safeHBoxes: 1,

allowDoubleClicks: 1,

showFontWarnings: 0,

method: "ProcessBeforeShowing",

loadFiles: ["plugins/mimeTeX.js","extensions/AMSsymbols.js"],

loadFonts: ["cmmib10"],

macros: {},

allowGlobal: 1,

noImageFonts: 0
```

Výpis 5: Část scriptu load.js

6.4 Script jsMath

Jak už jsem se jednou zmínil, je poněkud nešikovné zadávat matematiku pomocí párových tagů SPAN a DIV do HTML dokumentu, zvláště pro ty, kteří jsou zvyklí uvozovat matematiku latexovským způsobem. První způsob ulehčení bylo použití doplňku load.js, kde jsme nastavili hodnoty přímo ve scriptu. Při načtení klasického scriptu jsMath.js do stránek nám to samé umožňuje použití doplňku tex2math.js, který má automaticky nastaven všechny latexovské uvození. Problémem tohoto způsobu je však nesnadná implementace do zdrojového kódu HTML stránek, jelikož se do něj musí psát nejenom příkaz k uvození jednoho scriptu, ale i jeho nastavení, a to v pořadí určeném jsMath. V mých stránkách je použit onen těžší způsob. Jestliže požadujeme jiné nastavení než uvedené, měli bychom si řádně nastudovat materiál na domovských stránkách jsMath. Ze všech dostupných funkcí jsMath jsem použil překlad matematiky bez uvozování tagy SPAN a DIV, zvětšení matematického textu vůči okolí a nezobrazování varovného nápisu při nenainstalování LaTex fontů. Na následujícím výpise je požadované nastavení v HTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD␣XHTML␣1.0␣Strict//EN"
  "http :// www.w3.org/TR/2000/REC-xhtml1-20000126/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head><!-- ščřžýŠČŘŽÝ -->
   < title >nějaký title</ title >
   <meta http-equiv="content-type" content="text/html;␣charset=windows-1250" />
   <meta name="description" CONTENT="nějaký␣název␣">
   <meta name="author" CONTENT="nějaký␣autor">
   <link rel="stylesheet" type="text/css" href="cesta-k-stylum/styl.css" media="screen"/>
   <script> jsMath = {Controls: {cookie: {scale: 125}}} </script>
   <script src="cesta-k-jsMath/jsMath.js"></script>
   <script> jsMath.Setup.Script("plugins/tex2math.js") </script>
   <style> #jsMath_Warning {display: none} </style>
</head>

</body>

   Nějaký text stránky a $nějaká matematika$


   <script>
      jsMath.ConvertTeX();
      jsMath.Process(document);
   </script>

</body>

</html>
```

Výpis 6: Použití scriptu jsMath.js a doplňku tex2math

7 Ukázky příkladů

V této kapitole si uvedeme vždy jeden ukázkový příklad týkající se dané problematiky způsobu výpočtu integrálů. U všech zde uvedených příkladů jsou vypsány příkazy k vykreslení grafů. Interaktivní grafy jsou stejně jako zbytek sbírky příkladů s teorií použitou při výpočtu umístěny na webových stránkách `http://am.vsb.cz/milota`.

7.1 Dvojný integrál - použití fubiniovy věty

Příklad 7.1
Vypočtěte $\iint_M \frac{2y}{(1+x+y^2)^2}\mathrm{d}x\mathrm{d}y$, kde množina $M = \langle 0,1\rangle \times \langle 0,3\rangle$.

Řešení : Těleso T, které je shora ohraničeno funkcí $f(x,y) = \frac{2y}{(1+x+y^2)^2}$ na množině M a samotnou množinu M můžeme vykreslit pomocí příkazů :

```
> dzdxdyplot(z = 0..2*y/(1+x+y^2)^2, x = 0..1, y = 0..3, axes = boxed,
labels = [x, y, z], style = surfacewireframe, lightmodel = light2, glossiness
= .5, shading = zhue, orientation = [40, 60], scaling = constrained);
> dydxplot(y = 0..3, x = 0..1, labels = [x, y], axes = BOXED, scaling
= constrained);
```

Množina M je určena intervalem, což nám značně ulehčuje určení mezí. Jestliže uvažujeme množinu M jako elementární oblast vzhledem k x, jsou integrační meze :

$$0 \leq x \leq 1$$

$$0 \leq y \leq 3$$

$$
\begin{aligned}
I &= \iint_M \frac{2y}{\left(1+x+y^2\right)^2}\mathrm{d}x\mathrm{d}y = \int_0^1 \left(\int_0^3 \frac{2y}{\left(1+x+y^2\right)^2}\mathrm{d}y\right)\mathrm{d}x = \\
&= \int_0^1 \left[-\frac{1}{\left(1+x+y^2\right)}\right]_0^3 \mathrm{d}x = \int_0^1 \left(-\frac{1}{10+x}+\frac{1}{1+x}\right)\mathrm{d}x = \\
&= \int_0^1 \left(-\frac{1}{10+x}\right)\mathrm{d}x + \int_0^1 \left(\frac{1}{1+x}\right)\mathrm{d}x = \\
&= \left|\begin{matrix} 10+x=u \\ \mathrm{d}x = \mathrm{d}u \end{matrix}\right| = \\
&= -\int_{10}^{11}\left(\frac{1}{u}\right)\mathrm{d}u + \int_0^1 \left(\frac{1}{1+x}\right)\mathrm{d}x = -\ln(11)+\ln(10)+\int_0^1\left(\frac{1}{1+x}\right)\mathrm{d}x = \\
&= \left|\begin{matrix} 1+x=u \\ \mathrm{d}x = \mathrm{d}u \end{matrix}\right| = \\
&= -\ln(11)+\ln(10)+\int_1^2\left(\frac{1}{u}\right)\mathrm{d}u = -\ln(11)+\ln(10)+\ln(2).
\end{aligned}
$$

Nakonec si zobrazíme, pro lepší pochopení výpočtu integrálu, aproximaci funkce na dané množině.

```
> ApproximateInt(2*y/(1+x+y^2)^2, x = 0..1, y = 0..3, method = midpoint,
axes = boxed, orientation = [40, 60], coordinates = cartesian, partition
= [5, 5], output = plot, title = "", functionoptions = [color = "Navy"],
prismoptions = [color = "Coral"], scaling = unconstrained, labels = [x,
y, z]);
> ApproximateInt(2*y/(1+x+y^2)^2, x = 0..1, y = 0..3, method = midpoint,
axes = boxed, orientation = [40, 60], coordinates = cartesian, partition
= [10, 10], output = plot, title = "", functionoptions = [color = "Navy"],
prismoptions = [color = "Coral"], scaling = unconstrained, labels = [x,
y, z]);
> ApproximateInt(2*y/(1+x+y^2)^2, x = 0..1, y = 0..3, method = midpoint,
axes = boxed, orientation = [40, 60], coordinates = cartesian, partition
= [15, 15], output = plot, title = "", functionoptions = [color = "Navy"],
prismoptions = [color = "Coral"], scaling = unconstrained, labels = [x,
y, z]);
```

7.2 Dvojný integrál - transformace do polárních souřadnic

Příklad 7.2
Vypočtěte $\iint_M \sqrt{x^2+y^2}\mathrm{d}x\mathrm{d}y$, kde množina $M = \{(x,y) \in \mathbb{R}^2, 1 \leq x^2+y^2 \leq 4, x \leq y \leq 3\sqrt{x}\}$.

Řešení : Z nerovnice $1 \leq x^2+y^2 \leq 4$ je zřejmé, že jde o mezikruží kruhů se středy v bodě $[0,0]$ a poloměry 1 a 2. Dále je toto mezikruží omezeno nerovnicí $x \leq y \leq 3\sqrt{x}$. Integrační obor M je "výseč"kruhu. Proto použijeme substituci do polárních souřadnic. Tato substituce převádí integraci přes kruh na integraci přes dvojrozměrný interval. Integrační obor M před a po substituci si můžeme zobrazit příkazy :

```
> drdtplot(r = 1..2, theta = (1/3)*Pi..(1/4)*Pi, labels = [x, y], axes
= BOXED, title = "", color = black, scaling = constrained);
> dydxplot(y = 1 .. 2, x = (1/4)*Pi..(1/3)*Pi, labels = [t, r], axes
= BOXED, title = "", color = black, scaling = constrained);
```

Nakonec ještě můžeme srovnat tělesa před a po transformaci :

```
> dzdrdtplot(z = 0..sqrt(r^2*cos(theta)*cos(theta)+r^2*sin(theta)*sin
(theta)), r = 1..2, theta = (1/3)*Pi..(1/4)*Pi, axes = boxed, labels =
[x, y, z], style = surfacewireframe, lightmodel = light2, glossiness =
.5, shading = zhue, orientation = [280, 60], scaling = unconstrained);
> dzdxdyplot(z = 0..sqrt(x^2*cos(y)*cos(y)+x^2*sin(y)*sin(y)), x = (1/4)
*Pi..(1/3)*Pi, y = 1..2, axes = boxed, labels = [r, t, z], style = surface
wireframe, lightmodel = light2, glossiness = .5, shading = zhue, orientation
= [280, 60], scaling = constrained);
```

Substituce : $x = r\cos(t)$, $y = r\sin(t)$,

$$\mathcal{J}(r,t) = r.$$

Dosadíme do :

$$1 \leq x^2 + y^2 \leq 4,$$
$$1 \leq r^2 \cos^2(t) + r^2 \sin^2(t) \leq 4,$$
$$1 \leq r^2 \leq 4,$$
$$r \in < 1, 2 >,$$

Pro t máme omezující podmínky,

$$r \cos(t) \leq r \sin(t) \leq \sqrt{3} r \cos(t),$$
$$1 \leq \tan(t) \leq \sqrt{3},$$
$$t \in < \frac{\pi}{4}, \frac{\pi}{3} >,$$

$$\begin{aligned}
I &= \iint_M \sqrt{x^2+y^2}\mathrm{d}x\mathrm{d}y = \\
&= \int_{\frac{\pi}{4}}^{\frac{\pi}{3}} \left(\int_1^2 \left(\sqrt{r^2\cos^2(t) + r^2 \sin^2(t)} r \right) \mathrm{d}r \right) \mathrm{d}t = \\
&= \int_{\frac{\pi}{4}}^{\frac{\pi}{3}} \left(\int_1^2 \left(r^2 \right) \mathrm{d}r \right) \mathrm{d}t = \int_{\frac{\pi}{4}}^{\frac{\pi}{3}} \left(\frac{1}{3} \left[r^3 \right]_1^2 \right) \mathrm{d}t = \int_{\frac{\pi}{4}}^{\frac{\pi}{3}} \left(\frac{7}{3} \right) \mathrm{d}t = \\
&= \frac{7}{3} \left[t \right]_{\frac{\pi}{4}}^{\frac{\pi}{3}} = \frac{7}{3} \left(\frac{\pi}{3} - \frac{\pi}{4} \right) = \frac{7\pi}{36}.
\end{aligned}$$

■

7.3 Trojný integrál - použití fubiniovy věty

Příklad 7.3
Vypočtěte objem omezené a uzavřené množiny $M \in \mathbb{R}^3$, je-li M ohraničena plochami o rovnicích $|x| = 1$, $|y| = 1$, $0 \leq z \leq x^2 + y^2$.

Řešení :

- ⋄ $|x| = 1$ (roviny)
- ⋄ $|y| = 1$ (roviny)
- ⋄ $z = 0$ (rovina)
- ⋄ $z = x^2 + y^2$ (rotační parabolická plocha)

Integrační obor M a průmět do roviny xy můžeme zobrazit pomocí příkazů :

```
> dzdydxplot(z = 0..x^2+y^2, y = -1..1, x = -1..1, axes = boxed, labels
= [x, y, z], style = surfacewireframe, lightmodel = light2, glossiness
= .5, shading = zhue, orientation = [125, 60], scaling = unconstrained);
> dydxplot(y = -1..1, x = -1..1, labels = [x, y], axes = BOXED, title
= "", color = black, scaling = constrained);
```

Integrační meze tedy jsou :

$$-1 \leq x \leq 1$$

$$-1 \leq y \leq 1$$

$$0 \leq z \leq x^2 + y^2$$

$$\begin{aligned}
I &= \iiint_M 1\mathrm{d}x\mathrm{d}y\mathrm{d}z = \int_{-1}^{1}\left(\int_{-1}^{1}\left(\int_{0}^{x^2+y^2} 1\mathrm{d}z\right)\mathrm{d}y\right)\mathrm{d}x = \\
&= \int_{-1}^{1}\left(\int_{-1}^{1} x^2 + y^2\mathrm{d}y\right)\mathrm{d}x = \int_{-1}^{1}\left[x^2y + \frac{y^3}{3}\right]_{-1}^{1}\mathrm{d}x = \\
&= \int_{-1}^{1}\left(x^2 + x^2 + \frac{1}{3} + \frac{1}{3}\right)\mathrm{d}x = \left[\frac{2x^3}{3} + \frac{2x}{3}\right]_{-1}^{1} = \\
&= \frac{2}{3} + \frac{2}{3} + \frac{2}{3} + \frac{2}{3} = \frac{8}{3}.
\end{aligned}$$

■

7.4 Trojný integrál - transformace do válcových souřadnic

Příklad 7.4

Vypočtěte $\iiint_M xz\mathrm{d}x\mathrm{d}y\mathrm{d}z$, kde množina $M = \{(x, y, z) \in \mathbb{R}^3, x^2 + y^2 \leq 2x, x \leq z \leq 2x\}$.

Řešení :

Nejprve určíme mezní plochy :

- $\diamond$ $x^2 + y^2 = 2x$ (rotační válec)
- $\diamond$ $z = x$ (rovina)
- $\diamond$ $z = 2x$ (rovina)

Integrační obor M a průmět do roviny xy můžeme vykreslit za pomoci funkcí :

```
> dzdrdtplot(z = r*cos(theta)..2*r*cos(theta), r = 0..2*cos(theta), theta
= -(1/2)*Pi..(1/2)*Pi, axes = boxed, labels = [x, y, z], style = surface
wireframe, lightmodel = light2, glossiness = .5, shading = zhue, orientation
= [240, 60], scaling = constrained);
> drdtplot(r = 0..2*cos(theta), theta = -(1/2)*Pi..(1/2)*Pi, axes = boxed,
labels = [x, y], scaling = constrained);
```

Použijeme transformaci do válcových souřadnic.

Substituce : $x = r\cos(t)$, $y = r\sin(t)$, $z = z$,

$$\mathcal{J}(r, t, z) = r.$$

Dosazením do $x^2 + y^2 \leq 2x$ dostáváme :

$$r^2\cos^2(t) + r^2\sin^2(t) \leq 2r\cos(t),$$
$$r \leq 2\cos(t),$$
$$r \in < 0, 2\cos(t) >,$$

Pro t máme omezující podmínku,

$$2\cos(t) \geq 0,$$
$$t \in < -\frac{\pi}{2}, \frac{\pi}{2} >,$$

Pro z máme omezující podmínky,

$$x \leq z \leq 2x,$$
$$r\cos(t) \leq z \leq 2r\cos(t),$$
$$z \in < r\cos(t), 2r\cos(t) >,$$

Transformací do válcových souřadnic přejde množina M v množinu T a její průmět do roviny rt za použití

```
> dzdxdyplot(z = x*cos(y)..2*x*cos(y), x = 0..2*cos(y), y = -(1/2)*Pi..
(1/2)*Pi, axes = boxed, labels = [r, t, z], style = surfacewireframe,
lightmodel = light2, glossiness = .5, shading = zhue, orientation = [120,
60], scaling = constrained);
> dxdyplot(x = 0..2*cos(y), y = -(1/2)*Pi..(1/2)*Pi, axes = boxed, labels
= [r, t], color = black, scaling = constrained);
```

$$\begin{aligned}
I &= \iiint_M z\mathrm{d}x\mathrm{d}y\mathrm{d}xz = \\
&= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \left(\int_0^{2\cos(t)} \left(\int_{r\cos(t)}^{2r\cos(t)} \left(zr^2\cos(t) \right) \mathrm{d}z \right) \mathrm{d}r \right) \mathrm{d}t = \\
&= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \left(\int_0^{2\cos(t)} \left(r^2\cos(t) \left[\frac{z^2}{2} \right]_{r\cos(t)}^{2r\cos(t)} \right) \mathrm{d}r \right) \mathrm{d}t = \\
&= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \left(\int_0^{2\cos(t)} \left(r^2\cos(t) \frac{1}{2} \left(4r^2\cos^2(t) - r^2\cos^2(t) \right) \mathrm{d}r \right) \mathrm{d}t = \\
&= \frac{3}{2} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \left(\int_0^{2\cos(t)} \left(r^4\cos^3(t) \right) \mathrm{d}r \right) \mathrm{d}t = \frac{3}{10} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \left(\left[r^5 \right]_0^{2\cos(t)} \cos^3(t) \right) \mathrm{d}t = \\
&= \frac{3}{10} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} 32\cos^8(t)\mathrm{d}t = \frac{96}{10} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos^8(t)\mathrm{d}t = \\
&= \frac{96}{10}\frac{35\pi}{128} = \frac{21\pi}{8}.
\end{aligned}$$

Integrál $\int \cos^8(t)\mathrm{d}t$ se spočte pomocí goniometrického vzorce

$$\cos^2(t) = \frac{1}{2} + \frac{1}{2}\cos(2t).$$

Pro dlouhý a náročný výpočet zde vypočteno pomocí programu Maple. ■

7.5 Trojný integrál - transformace do sférických souřadnic

Příklad 7.5

Vypočtěte $\iiint_M \sqrt{x^2+y^2}\mathrm{d}x\mathrm{d}y\mathrm{d}z$, kde množina $M = \{(x,y,z) \in \mathbb{R}^3, x^2+y^2+z^2 \leq 4^2, \sqrt{x^2+y^2} \leq z\}$.

Řešení :

Nejprve určíme mezní plochy :

$\diamond$ $x^2+y^2+z^2 = 4^2$ (kulová plocha)

$\diamond$ $\sqrt{x^2+y^2} = z$ (parabolická plocha)

Integrační obor M a průmět do roviny xy zobrazíme použitím funkcí :

```
>  dpdtdphiplot(rho = 0..4, theta = 0..2*Pi, phi = 0..(1/4)*Pi, axes =
boxed, labels = [x, y, z], style = surfacewireframe, lightmodel = light2,
glossiness = .5, shading = zhue, orientation = [100, 60], scaling = constrained);
>  drdtplot(r = 0..4, theta = 0..2*Pi, axes = boxed, labels = [x, y],
scaling = constrained);
```

Použijeme transformaci do sférických souřadnic.

Substituce : $x = \rho\cos(\phi)\cos(\psi)$, $y = \rho\sin(\phi)\cos(\psi)$, $z = \rho\sin(\psi)$,

$$\mathcal{J}(\rho,\phi,\psi) = \rho^2\cos(\psi).$$

Dosadíme do rovnice $x^2+y^2+z^2 \leq 4^2$ a dostaneme :

$$\rho^2\cos^2(\phi)\cos^2(\psi) + \rho^2\sin^2(\phi)\cos^2(\psi) + \rho^2\sin^2(\psi) \leq 4^2,$$

$$\rho^2 \leq 4^2,$$

$$\rho \leq 4,$$

$$\rho \in < 0, 4 >,$$

Pro ψ máme omezující podmínku :

$$\sqrt{\rho^2\cos^2(\phi)\cos^2(\psi) + \rho^2\sin^2(\phi)\cos^2(\psi)} \leq \rho\sin(\psi),$$

$$\tan(\psi) \geq 1,$$
$$\psi \in < \frac{\pi}{4}, \frac{\pi}{2} >,$$

Pro ϕ nemáme žádnou omezující podmínku,

$$\phi \in < -\pi, \pi >,$$

Transformací do sférických souřadnic přejde množina M v množinu T s průmětem do roviny $\phi\psi$, který můžeme zobrazit funkcemi :

```
> dzdydxplot(z = 0..4, y = 0..2*Pi, x = (1/4)*Pi..(1/2)*Pi, axes = boxed,
labels = [psi, phi, rho], style = surfacewireframe, lightmodel = light2,
glossiness = .5, shading = zhue, orientation = [120, 60], scaling = constrained);
> dydxplot(y = 0..2*Pi, x = (1/4)*Pi..(1/2)*Pi, axes = boxed, labels
= [psi, phi], color = black, scaling = constrained);
```

$$I = \iiint_M \sqrt{x^2+y^2}\mathrm{d}x\mathrm{d}y\mathrm{d}z = \int_0^{2\pi} \left(\int_{\frac{\pi}{4}}^{\frac{\pi}{2}} \left(\int_0^4 \left(\rho^2 \cos(\psi)\rho \right) \mathrm{d}\rho \right) \mathrm{d}\psi \right) \mathrm{d}\phi =$$
$$= 32 \int_0^{2\pi} \left(\int_{\frac{\pi}{4}}^{\frac{\pi}{2}} \left(1 + \cos(2\psi) \right) \mathrm{d}\psi \right) \mathrm{d}\phi = 32 \int_0^{2\pi} \left[\psi + \frac{\sin(2\psi)}{2} \right]_{\frac{\pi}{4}}^{\frac{\pi}{2}} \mathrm{d}\phi =$$
$$= 32 \int_0^{2\pi} \left(\frac{\pi}{2} + 0 - \frac{\pi}{4} - \frac{1}{2} \right) \mathrm{d}\phi = 32 \int_0^{2\pi} \left(\frac{\pi}{4} - \frac{1}{2} \right) \mathrm{d}\phi = 64\pi \left(\frac{\pi}{4} - \frac{1}{2} \right) = 16\pi^2 - 32.$$

■

8 Závěr

Ve své práci jsem se zabýval propojením knihovny JavaViewLib s programem Maple za účelem prezentace problematiky dvojných a trojných integrálů na webu a vytvořil jsem interaktivní webovou galerii obsahující jak příklady, tak teorii potřebnou k jejímu výpočtu.

Domnívám se, že se mi podařilo vytvořit jednak stránky, které mohou sloužit jako doplňující materiál při výuce dvojných a trojných integrálu, jednak manuál, jenž shrnuje všechny kroky k vytvoření podobných interaktivních stránek s obdobným tématem. Většinu originálních anglických manuálů jsem přeložil do českého jazyka a odstranil jsem tak možná poslední překážku pro dychtivé jedince.

Myslím, že i implementace latexovskeho zápisu matematiky do HTML kódu pomocí jsMtah je výzvou k vytvoření pěknějších stránek s matematickým obsahem. Mnou vytvořená webová galerie je rozšiřitelná o další kapitoly zabývajícími se integrály jako například křivkový nebo plošný nebo mohou být vytvořeny interaktivní testy na probrané témata. Pro snadnou orientaci uživatele jsem stránky vytvořil co možná nejpřehledněji.

Tato práce by měla pomáhat uživatelům k seznámení se s popsanými programy. Dále by měla pomoci překonat prvotní nezdary způsobené chybami při zadávání příkazů. Z vlastní zkušenosti mohu říci, že maličkosti typu zaměnění čárky a středníku mohou vést k frustraci. Trpělivost a přesnost však zaručují kvalitní výsledky.

Text diplomové práce byl vysázen systémem LaTeX. Jedná se nadstavbu programu TeX, který umožňuje autorům textů sázet a tisknout dokumenty v nejvyšší možné typografické kvalitě a je vhodný zejména pro sazbu matematických textů.

Vojtěch Milota

9 Reference

[1] ŠARMANOVÁ, P., PLCH R.,: *Interaktivní prezentace matematické grafiky na webu a v PDF dokumentech.*, Sborník semináře Technologie pro e-vzdělávání 2007, Praha: katedra počítačů ČVUT FEL, 2007, 31-38, ISBN 978-80-01-03756-0.

[2] ŠARMANOVÁ, P., PLCH R.,: *Maple, JavaViewLib a JavaView - nástroje k tvorbě, exportu a prezentaci interaktivní 3D grafiky.*, Sborník 6. matematického workshopu, 2007, 101-102, FAST VUT Brno, ISBN 80-214-2741-8.

[3] ŠARMANOVÁ, P., PLCH R.,: *Interaktivní 3D grafika v HTML a PDF dokumentech.*, Zpravodaj Československého sdružení uživatelů TEXu, Praha, Československé sdružení uživatelů TEXu. 2008, vol. 18, no. 1-2, s. 76-92, ISSN 1211-6661 (tištěná verze), ISSN 1213-8185 (online verze).

[4] FILIPEC, Z., PLCH R.,: *Maple a JavaView.* Využití Maplu ve výuce a výzkumu na vysokých školách a akademiích věd., 1. vyd. Brno: Econ publishing s.r.o., 2002, ISBN 80-86433-10-2.

[5] ŠELLEOVÁ, S.: *Prezentace matematické grafiky na webu s programem Java View.*, diplomová práce MU Brno, 2007.

[6] MUSIL, V.: *Prezentace matematické grafiky (Integrální počet funkcí více proměnných) na webu s programem JavaView*, diplomová práce MU Brno, 2007.

[7] CHARVÁT, J. et al.: *Příklady k Matematice II.*, 1. vyd. Praha: ČVUT, 1993. 334 s. ISBN 80-01-00887-8.

[8] KUBEN, J., HOŠKOVÁ, Š., Račková P.: *Integrální počet funkcí více proměnných.* 1.vyd. Brno: Univerzita obrany, 2005. 146 s. ISBN 80-7231-031-3.

[9] *JavaView - Interactive 3D Geometry and Visualization*, duben 2009.Dostupné na www : `http://www.javaview.de/`.

[10] *jsMath - A Method of Including Mathematics in Web Pages*, duben 2009.Dostupné na www : `http://www.math.union.edu/~dpvc/jsMath/`.

[11] *Maple - Math Software for Engineers, Educators and Students*, duben 2009.Dostupné na www : `http://www.maplesoft.com/`.

[12] DRUSKA, P.: *CSS a XHTML tvorba dokonalých webových stránek krok za krokem.*, 1.vyd. Praha: Grada Publishing, a.s., 2006. 200 s. ISBN 80-247-1382-9.

[13] PROKOP, M.: *CSS kaskádové styly pro webdesignéry.*, 2. vyd. Brno: CP Books, a.s., 2005. 288 s. ISBN 80-251-0487-7.

[14] MACH, J.: *PHP pro úplné začátečníky.*, 1. Vyd. Brno: CP Books, a.s., 2005. 167 s. ISBN 80-7226-834-1.

[15] LOMTATIDZE, L., PLCH, R.: *Sázíme v LATEXU diplomovou práci z matematiky.*, 1. vyd. Brno: Masarykova univerzita, 2003. 122 s. ISBN 80-210-3228-6.